Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

Computer Programs for Investigating the Effects of Environmental Events on a Time Series of Recruitment

J.M. Hoenig, M.H. Prager,
and N.B. Payton

Science Branch
Department of Fisheries and Oceans
P.O. Box 5667
St. John's, Newfoundland A1C 5X1

November 1989

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1713

Fisheries and Oceans Pêches et Océans

Canada

Canadian Technical Report of
Fisheries and Aquatic Sciences

Rapport technique canadien des
sciences halieutiques et aquatiques

Canadian Technical Report of

Fisheries and Aquatic Sciences 1713

November 1989

COMPUTER PROGRAMS FOR INVESTIGATING THE EFFECTS OF ENVIRONMENTAL EVENTS ON A TIME SERIES OF RECRUITMENT

by

J. M. Hoenig, M. H. Prager[1], and N. B. Payton

Science Branch

Department of Fisheries and Oceans

P.O. Box 5667

St. John's, Newfoundland A1C 5X1

[1]Department of Oceanography, Old Dominion University, Norfolk, VA 23529, U.S.A.

©Minister of Supply and Services Canada 1989

Cat. No. Fs 97-6/1713E ISSN 0706-6457

Correct citation for this publication:

Hoenig, J. M., M. H. Prager, and N. B. Payton. 1989. Computer programs for investigating the effects of environmental events on a time series of recruitment. Can. Tech. Rep. Fish. Aquat. Sci. 1713: v + 43 p.

CONTENTS

Abstract/Résumé iv
Preface v
Introduction 1
Program EPOCH 2
    Description 2
    Program preparation 2
    Data input 3
    Program output 3
Program EPOWER 3
    Description 3
    Program preparation 4
    Data input 4
    Program output 4
Acknowledgments 5
References 5
Appendix I - Program EPOCH sample input file 6
Appendix II - Program EPOCH listing 7
Appendix III - Program EPOWER sample input file 20
Appendix IV - Program EPOWER listing 21

ABSTRACT

Hoenig, J. M., M. H. Prager, and N. B. Payton. 1989. Computer programs for investigating the effects of environmental events on a time series of recruitment. Can. Tech. Rep. Fish. Aquat. Sci. 1713: v + 43 p.

Prager and Hoenig (in press) used a randomization procedure to test the significance of discrete environmental events, such as el Niño-Southern Oscillation events, on a time series of recruitment. Their procedure implicitly accounted for autocorrelation and other problems frequently encountered when trying to use a parametric testing procedure. In a follov up paper, Prager and Hoenig (in prep.) investigated the pover of several randomization tests and compared them to parametric alternatives. This report presents the two Fortran computer programs used in these studies.

RÉSUMÉ

Hoenig, J. M., M. H. Prager, and N. B. Payton. 1989. Computer programs for investigating the effects of environmental events on a time series of recruitment. Can. Tech. Rep. Fish. Aquat. Sci. 1713: v + 43 p.

Prager et Hoenig (sous presse) ont employé une méthode de randomization afin de tester le degré de significance d'événements environementaux discrets, tels que les événements El Niño, sur une série chronologique de recrutement. Leur méthode tenait compte de façon implicite de l'autocorrélation ainsi que d'autres problèmes rencontrés fréquemment lors de l'emploi de méthodes de test paramétriques. Dans un autre article, Prager et Hoenig (en préparation) ont étudié la puissance de plusieurs tests de randomization et les ont comparés à d'autres tests paramétriques. Dans ce rapport on présente les deux programmes Fortran qui ont servi dans ces études.

PREFACE

This report provides descriptions, listings, and user's instructions for two Fortran computer programs used by Prager and Hoenig (in press; in review) to study environmental influences on recruitment. The purpose of the report is to document the computational procedures used and to ensure that the programs remain available to future investigators. We have tested these programs carefully. However, as with any large program, it is not possible to state categorically that the programs are error-free. Users are warned to use the programs at their own risk. The programs are available through the American Fisheries Society's Computer Users' Section.

INTRODUCTION

In recent years, a great deal of attention has been focused on the problem of determining to what extent environmental conditions influence recruitment. A large number of studies have involved the use of parametric statistical methods (e.g. regression, correlation, t-tests) to describe and test the relationship between environmental factors and recruitment in available time series. These approaches generally require the following assumptions: independence of observations (i.e., from year to year), normality, and homogeneity and/or homoscedasticity of variances. All of these assumptions may be suspect when dealing with time series data. In addition, the parametric methods assume random sampling (of time) which is obviously not possible. Few studies have considered the effects of failure to meet the necessary assumptions.

One alternative to the above-mentioned parametric procedures is to use classical time series methods as described, for example, by Box and Jenkins (1970). Prager and Hoenig (in press) described another approach based on randomization tests. We concentrate on the latter approach in this report.

The randomization approach of Prager and Hoenig (in press) is based on comparing the recruitment in key event years, i.e. years in which a specified type of environmental event occurred, with recruitment in background years, i.e. the years immediately surrounding the key event years. For example, key event years might be taken to be years in which an el Niño-Southern Oscillation event occurred or years in which offshore transport in May exceeded a pre-specified threshold. Background years might be defined as the two years before and the two years after a key event year. Alternatively, one might define background years as the two years before a key event year and ignore the years immediately after the key events. (This would be appropriate if one had reason to believe that the effect of the key event carried over into the years immediately after the event year.) In the parlance of Haurwitz and Brier (1981), a key event year and its associated background year together comprise an epoch. Superposing (i.e. averaging) a set of epochs provides a visual method for assessing the "average" trend over the course of epochs.

The goal, then, is to test whether recruitment in key event years tends to be different from that in background years. The usual (parametric) approach to comparing two means is to compute some sort of t-statistic and refer to a table of Student's t-distribution to determine the significance level. In the case considered here, we could certainly compute a t-statistic. However, we would be ill-advised to assume this statistic follows a Student's-t under the null hypothesis because of failures of the assumptions. However, we can determine the null distribution, and hence the significance level, by casting the problem as a randomization test (see below). In this way we implicitly account for the failures of assumptions associated with the parametric approach.

For a randomization test, we assume that the available observations comprise the entire population of interest, i.e. that we are not dealing with a sample from some larger population. We then test whether the association between two variables could have arisen by the chance "assignment" of the values of each variable to the units in the population. For example, the units in the population could be years; the values of one variable, recruitment, could have been assigned to the units according to some arbitrary process. Then the question becomes: could the label "key event" (versus "not-key event") have been assigned to the units (years) independently of the recruitment variable? Or, is there evidence that the key events have some association with the recruitment values, e.g. tending to occur when recruitment is high?

Suppose we wish to measure the strength of the association between recruitment and key event years by computing a t-statistic (exactly as one would for a Student's t-test). We need to determine the significance of the computed statistic with respect to the null hypothesis (that recruitment and key event status [yes or no] are assigned independently to years). Under the null hypothesis, the label "key event year" has no significance with respect to recruitment. Hence, key events could occur anywhere with respect to recruitment. Consequently, the exact null distribution of the test statistic could be determined by computing the value of the test statistic for every possible allocation of key events to years. For example, if 4 key events occurred in a 30 year period, there would be 30 choose 4 = 27,405 possible allocations of key events to years. Since the number of allocations can be large, an attractive alternative, which we used, is to determine the approximate null distribution by a Monte Carlo method. In this approximation version, a given number of "trials" is generated. For each trial, one of the possible allocations (of key events to year) is randomly selected. (The recruitment time series remains as originally observed.) The resulting histogram of computed values of the test statistic approximates the null distribution. If the observed value of the test statistic occurs in the extreme tail of the null distribution, the null hypothesis is rejected.

The interested reader is referred to Edgington (1987) for further information on randomization tests. Details of our use of randomization tests for testing hypotheses about recruitment are presented in Prager and Hoenig (in press).

Program EPOCH can be used to test for an association between discrete environmental events and a time series of recruitment (or relative recruitment or productivity). Since there are various ways to describe the degree of association (i.e. various possible test statistics), the program can be used to determine the significance level for three test statistics.

Program EPOWER was written to investigate the power of the statistical test under various assumed conditions (see Prager and Hoenig in prep.). Power is defined here as the probability of rejecting the null hypothesis under specified conditions. The program can be used to answer two kinds of questions:

1) which test statistics are likely to be the most powerful for a given type of recruitment pattern; and

2) what kind of power might be expected in practice if recruitment follows a given process and key events affect recruitment in a given way.

It is difficult to determine analytically which test statistics are the most powerful and what power levels can be achieved. Program EPOWER is written in a modular style so that additional test statistics and additional patterns of recruitment can be added to the program as needed.

PROGRAM EPOCH

DESCRIPTION

This Fortran 77 program reads in a time series of recruitment and a list of key event years. It also reads in the (user-specified) definition of background years (either the $\underline{k}$ years before each key event or the $\underline{k}$ years before and the $\underline{k}$ years after each key event), and the number of Monte Carlo trials desired. Three test statistics can be computed (D, T, or W - see below). The program then estimates the significance level for the chosen test statistic by a Monte Carlo randomization test. The significance level is estimated by $(x + 1)/(v + 1)$ where v is the number of Monte Carlo trials and x is the number of trials in which the computed test statistic exceeds the value of the test statistic for the actual data.

The simplest test statistic, D, is the difference between the mean recruitment in key event years ($\bar{E}$) and the mean value in background years ($\bar{B}$):

$$D = \bar{E} - \bar{B}.$$

The second statistic, T, was modified from Haurwitz and Brier (1981). It is computed as one would compute an ordinary Student's t-statistic:

$$T = (\bar{E} - \bar{B}) / S,$$

where $\bar{E}$ and $\bar{B}$ are as defined for the D statistic. The pooled standard deviation S is computed as for a t-test:

$$S = \left[\frac{(SS_E + SS_B)(N_E + N_B)}{N_E N_B (N_E + N_B - 2)} \right]^{1/2},$$

where SS_E is the sum of squared deviations of the N_E values of recruitment in key event years from their mean:

$$SS_E = \sum_{i=1}^{N_E} (E_i - \bar{E})^2,$$

where E_i is the recruitment in key event year i. Similarly, SS_B is the sum of the squared deviations of the N_B recruitment values in background years from their grand mean.

The W statistic is analogous to Student's t-statistic computed for paired data. Each key event year is compared with its own background years:

$$W = \bar{d} \sqrt{N_E} / S_w,$$

where S_w is a measure of dispersion (defined below), N_E is as defined for the T statistic, and $\bar{d}$ is the mean of all paired differences between the recruitment value in key event year i and the recruitment value in each corresponding background year. More precisely,

$$\bar{d} = \frac{1}{N_B} \sum_{i=1}^{N_E} \sum_{j=1}^{n_i} (E_i - B_{ij}),$$

where n_i is the number of background values available on recruitment for key event i. Generally n_i is a constant (e.g. 2 years before + 2 years after = 4), but it has fewer observations when the key event occurs near a missing value of the response or near either end of the time series. In other words, missing values of recruitment are accommodated by eliminating them from computation of the test statistic, and adjusting n_i and N_B accordingly. The measure of dispersion S_w is computed as for a paired t-test:

$$S_w = \left[\frac{1}{N_B - 1} \sum_{i=1}^{N_E} \sum_{j=1}^{n_i} (E_i - B_{ij} - \bar{d})^2 \right]^{1/2}.$$

Output from the program consists of two files. One contains information for plotting estimated significance level as a function of the number of Monte Carlo trials. In this way, one can examine whether the estimated significance level (P-value) has stabilized. The other output file documents the data and test options used, and presents the test results.

Program EPOCH uses a slightly modified version of subroutine RAN3 (© 1986 by Numerical Recipes Software. Reproduced by permission, from the book Numerical Recipes: The Art of Scientific Computing, published by Cambridge University Press).

PROGRAM PREPARATION

The program is written in standard Fortran 77 and could easily be run on a microcomputer. The program operates in batch mode - all input is read from a single data file called EPOCH.DAT and output is written to two files. One output file contains information for plotting estimated significance level as a function of the number of Monte Carlo trials (to determine whether the estimated significance level has stabilized). The second output file contains a summary of the data

and parameters used in the test and the test results.

The data arrays are dimensioned to allow up to 100 observations (years of data); 50 key events, and to permit up to 10,000 Monte Carlo trials. If this arrangement is satisfactory the program can be compiled and run without modification.

DATA INPUT

A sample data input file is presented in Appendix I. Parameters and data for analysis are entered into a data file called EPOCH.DAT in free format, as follows:

line 1: number of years (rows) of recruitment data (= NYear)
line 2: number of Monte Carlo trials
line 3: number of key events (= NKey)
next NKey rows: one pair of numbers in each row, each row pertaining to a key event. First number = Year ID for the key event; second number = indicator for expected response (see (1) below)
next line: minimum spacing for key events (see (2) below)
next line: indicator for which test statistic should be used (enter 'T', 'D', or 'W' [n single quotes])
next line: seed for random number generator RAN3 from Numerical Recipes (enter a negative integer)
next line: epoch definition - enter 1 if only background years preceding the key events are used; enter 2 if epoch is two-sided, i.e. background years on each side of the key events are used.
next line: number of background years preceding each key event (and also succeeding each key event year, if two-sided epochs are used).
next NYear lines: time series of recruitment, one year per line. Each line consists of two numbers - the year ID and the recruitment value. Years for which no recruitment data are available are left out of the file.

Notes:

(1) Enter +1 if the recruitment is hypothesized to be higher than background (i.e. a "peak") for the corresponding type of key event (under the alternative hypothesis); enter -1 if recruitment is hypothesized to be lower than background (i.e. a trough). A combination of pluses and minuses is used for reflected event analysis.

(2) Minimum spacing for key events - there may be physical or other reasons for believing that key events cannot occur in adjacent years. In this case, the randomizations (Monte Carlo trials) can be generated subject to the restriction that key events have a minimum spacing. Otherwise enter a 1.

PROGRAM OUTPUT

File EPOCH.SIG contains information on the estimated significance level as a function of the number of Monte Carlo trials. A two line header (label) is followed by two blank lines. The remaining lines give, for every second Monte Carlo trial, the following information on a line: Monte Carlo trial number i, value of the test statistic for the trial data i, and running significance level ([x+1]/[i+1], where x is the number of trial data sets for which the test statistic is greater than that observed for the actual data!. The estimated significance level can be plotted against the trial number to see if the estimated significance has stabilized.

File EPOCH.OUT documents the analysis with the following information:

width of epoch
# of background years preceding the key events
# of background years following the key events
the random number generator seed
rows of input data
the year ID for the first row of data
the year ID for the last row of data
range of ID values
# of rows with missing data
# of key events (NKey)
the next NKey rows list the year ID of the key event years and the expected direction of hs difference between key event years and background years (i.e., positive or negative)
which test statistic was used
value of the test statistic for the actual data
# of Monte Carlo trials
minimum spacing allowed between the key events in the Monte Carlo trials
number of Monte Carlo trials with test statistic greater than that observed for the actual data
estimated significance level

PROGRAM EPOWER

DESCRIPTION

This Fortran 77 program estimates the realized alpha level and the power of the randomization test under various possible conditions. The simulated conditions can be such that a parametric test (e.g. a t-test) is appropriate. That is, the observations in the time series can be generated independently from a normal distribution with observations in key event years being merely shifted by some amount delta. Or, the time series can be generated with a linear trend or with autocorrelation (AR1 model) or both.

In general, recruitment value in year i is generated according to the general model

$$XX_i = \text{slope} * i + AR * XX_{i-1} + \text{delta} * Z_i + e_i$$

where XX_i is the recruitment value in year i, slope is the trend (per year) in recruitment, AR is the autoregressive parameter, delta is the amount by which key event years differ from non-key event years, Z_i is an indicator variable for key event status (=1 if year i is a key event year), and e_i is a standard normal random variable.

Six test statistics/test procedures are included in the program. Two of these are the usual parametric tests: t-test and paired t-test, with signficance level determined by referring to

Student's t-distribution. The other test procedures involve determining the significance level by randomization. The program can be used to determine the penalty for using a randomization test when a parametric test is appropriate and vice versa.

The test statistics are as follows:

1) $D = \bar{E} - \bar{B}$ = difference of means; see detailed description in writeup for Program EPOCH. Significance is determined by randomization.

2) $T = (\bar{E} - \bar{B}) / S$ = parametric t-statistic; see detailed description in writeup for program EPOCH. Significance is determined by referring to a table of Student's t-distribution with $N_E + N_B - 2$ degrees of freedom where N_E and N_B are the number of observations on key event and background years, respectively.

3) T_R = same as (2) but significance is determined by randomization.

4) $W = \bar{d} \sqrt{N_B}/S_W$ = paired t-statistic; see detailed description in writeup for program EPOCH. Significance is determined by randomization.

5) Q = paired t-statisic defined as follows:

$$Q = \frac{\bar{D}\sqrt{N_B}}{S_Q}$$

where

$$\bar{D} = \sum_{i=1}^{N_E} (E_i - \bar{B}_i)$$

$$S_Q = \left[\frac{\sum_{i=1}^{N_E} (E_i - \bar{B}_i - \bar{D})^2}{N_E - 1} \right]^{\frac{1}{2}}$$

and N_E is the number of key event years, and $\bar{B}_i$ is the mean of the recruitment values for background years associated with key event i. Significance is determined by referring to a table of Student's t-distribution with $N_E - 1$ degrees of freedom.

6) Q_R = same as (5) but significance is determined by randomization.

Program EPOWER uses slightly modified versions of the following routines: RAN3, GASDEV, AVEVAR, TTEST, TPTEST, BETAI, BETACF, GAMMLN. (© 1986 by Numercial Recipes Software. Reproduced by permission, from the book Numerical Recipes: The Art of Scientific Computing, published by Cambridge University Press.)

PROGRAM PREPARATION

The program is written in standard Fortran 77. Although memory requirements are small, so the program could be run on a microcomputer, the execution time is substantial. We used about 100 hours of CPU time on a mainframe supercomputer to generate the series of power curves in Prager and Hoenig (in prep). It took approximately 20 minutes of CPU time to estimate the power for one set of conditions.

The program operates in batch mode - all input is read from a single data file called EPOWER.DAT and output is written to two files. One file, EPOWER.OUT, provides for each test statistic the following information: number of simulated data sets for which a significant result was found, the estimated power, and ar approximate 95% confidence interval (based on the normal approximation to a binominal).

The other output file, EPOWER.DET, provides more detailed output (DET for detail). The file contains the value of the test statistic, and the estimated significance level, for every data set generated.

The program's arrays are dimensioned to allow up to 10 key event years to occur in a 50 year time series. If this is satisfactory, the program can be compiled and run without modification.

DATA INPUT

A sample data input file is presented in Appendix III. Parameters of the simulation are read in free format from a data file called EPOWER.DAT, as follows:

line 1: Number of years in each simulated data set.
line 2: Number of key event years in each data set.
line 3: Amount of difference in mean recruitment between background and key event years.
line 4: Slope of (trend in) recruitment over time (yr^{-1}).
line 5: Seed for random number generator RAN3 (from Numerical Recipes). Number should be negative.
line 6: Enter a 1 if only years before each key event are used as background years; enter a 2 if years before and after are used as background years.
line 7: Number of background years proceeding each key event year (and also succeeding each key event year, if two-sided epochs are used).
line 8: Number of data sets generated (outer trials - to be analyzed by the Monte Carlo randomization test)
line 9: Number of Monte Carlo randomizations (inner trials - for analyzing a given data set)
line 10: Autoregressive parameter

PROGRAM OUTPUT

File EPOWER.OUT provides a summary of the simulated conditions and the estimated power

levels with approximate 95% confidence limits. The information is provided as follows:

title ("Power analysis ....")
number of data sets generated randomly
number of years in each data set generated
number of key event years in each data set
definition of an epoch
model type (autoregressive vs. uncorrelated)
delta, the amount added to the recruitment value to make the year a key event year
slope of trend: the amount by which the expected recruitment changes per year
autoregressive parameter for the AR1 model
number of Monte Carlo randomizations for a given generated data set
random number seed for RAN3
title ("Summary of Results")
Statistics
number of significant results (rejections of H_0)
estimated power
lower 95% confidence limit
upper 95% confidence limit

The detailed output file, EPOWER.DET, provides the exact same summary of simulated conditions as EPOWER.OUT. It then provides, for every simulated data set, the value of the statistics and their estimated significance levels.

ACKNOWLEDGMENTS

We thank the Computer Center and the Department of Oceanography at Old Dominion University for providing computer facilities and support for program development. We especially thank William Vetterling and Numerical Recipes Software for permission to reproduce several proprietary software routines. Marianne Eckenswiller and Janice Lannon typed the manuscript.

REFERENCES

Box, G. E. P. and G. M. Jenkins. 1970. Time series analysis-forecasting and control. Holden-Day, San Francisco, CA.

Edgington, E. S. 1987. Randomization tests, second edition. Marcel Dekker, New York.

Haurwitz, M. W. and G. W. Brier. 1981. A critique of the superposed epoch analysis method: its application to solar-weather relations. Monthly Weather Review 109: 2074-2079.

Prager, M. H. and J. M. Hoenig. In press. Superposed epoch analysis: a randomization test of environmental effects on recruitment, with application to chub mackerel. Trans. Amer. Fish. Soc. 118 (6).

Prager, M. H. and J. M. Hoenig. In prep. Can we determine the significance of key events on time series of recruitment? - a power analysis. To be submitted to Trans. Amer. Fish. Soc.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1986. Numerical recipes - the art of scientific computing. Cambridge Univ. Press, New York.

Appendix I - Program EPOCH
sample input file (EPOCH.DAT) with annotations

40	Number of years (rows) of data.
10000	# Monte Carlo trials.
8	# of key events.
1933 -1	Key event year ID's and expected directions of
1941 +1	difference between key event year and background years.
1951 +1	
1955 -1	
1957 +1	
1958 +1	
1959 +1	
1964 -1	
1	Minimum spacing for key events.
'W'	Test statistic used.
-12345	Seed for random # generator (negative integer).
2	Epoch definition: 1 for background years to the left. 2 for left & right.
2	Number of background years preceding (and possibly following) key events.
1929 0.701664	Year ID, recruitment value.
1930 0.738894	
1931 0.447222	
1932 0.229122	
1933 0.127301	
1934 0.077382	
1935 0.124930	
1936 0.363709	
1937 0.388174	
1938 0.671057	
1939 0.530253	
1940 0.736018	
1941 1.416940	
1942 0.655209	
1943 0.400316	
1944 0.338645	
1945 0.160253	
1946 0.205481	
1947 2.403677	
1948 2.454197	
1949 0.154710	
1950 0.040119	
1951 0.052797	
1952 1.682868	
1953 10.02919	
1954 1.189460	
1955 1.132808	
1956 0.211062	
1957 0.362257	
1958 2.482832	
1959 2.089239	
1960 2.041735	
1961 1.052638	
1962 0.136613	
1963 0.056806	
1964 0.023837	
1965 0.095131	
1966 0.0326	
1967 0.2775	
1968 1.5857	

Appendix II - Program EPOCH Listing

```
C ************************************************************************      0001
      PROGRAM EPOCH                                                           0002
C ************************************************************************      0003
C     Prager/Hoenig/Payton  November, 1989                                    0004
C                                                                             0005
C     Use superposed epoch analysis to test hypotheses.                       0006
C     This program uses an epoch of specified half-width on one side,         0007
C     or centered around, the "key event."  It also expects to see            0008
C     the resulting effect in the same year as the key event.                 0009
C     Reference:  Prager, M. H. and J. M. Hoenig. (in press).                 0010
C                 Superposed epoch analysis: a randomization test of          0011
C                 enviromental effects on recruitment, with application       0012
C                 to chub mackerel. Trans. Amer. Fish. Soc. 118(6)            0013
C ---------------------------------------------------------------------       0014
C     Input Data File Structure (Free Format):                                0015
C     ----------------------------------------                                0016
C     Line 1:      NYear     Number of years (rows) in data matrix.           0017
C     Line 2:      NSim      Number of Monte Carlo trials.                    0018
C     Line 3:      NKey      Number of key events.                            0019
C     Lines 4 to                                                              0020
C     NKey+3:      Pairs of numbers, one pair for each key event.           0021
C                  Key()     Year ID's of key events.                       0022
C                  ISign():  +1 for key event(peak), -1 for reflected       0023
C                            key event (trough).                            0024
C     Next Line:   MinDif    Minimum spacing of key events.                   0025
C     Next Line:   AStat     Flag for statistic to be used.                   0026
C                            'T': T-statistic.                                0027
C                            'D': D-statistic.                                0028
C                            'W': W-statistic.                                0029
C     Next Line:   MSeed     Negative seed for RAN3                           0030
C     Next Line:   NSide     Sides to be used for MWidth.                     0031
C                            '1': One sided. Uses left side.                  0032
C                            '2': Two sided.                                  0033
C     Next Line:   MWidth    Half-width of epoch on each side of              0034
C                            key event if two sided, full-width if one        0035
C                            sided.                                           0036
C     Next NYear Lines:                                                       0037
C                  pairs of IYear() and XX() values.                          0038
C                  IYear():  Year ID's of each row of data matrix             0039
C                  XX():     Dependent variable of each row                   0040
C ---------------------------------------------------------------------       0041
C  ***  Units & ddnames of files  ***                                         0042
C                                                                             0043
C  Unit      DDname      Action   Contents                                    0044
C  -------   ----------  ------   ------------------------------              0045
C  Unit 12   epoch.dat   read     Data                                        0046
C  Unit 13   epoch.out   write    Main Output File                            0047
C  Unit 14   epoch.sig   write    Series of P-values for plotting             0048
C ---------------------------------------------------------------------       0049
C  ***  Major variables  ***                                                 0050
C                                                                             0051
C  IYear()         Year ID's (col. 1 of input data matrix)                    0052
C  XX()            Dependent variable (col. 2 of input data matrix)           0053
C  NYear           Number of years (rows) in the input data matrix            0054
C  NKey            Number of key events                                       0055
C  Key()           Year ID's of key events (input)                            0056
C  Key2()          Year ID's of key events (Monte Carlo)                      0057
C  NSim            Number of Monte Carlo trials                               0058
C  ILowYr          Value of IYear(1)                                          0059
C  IHiYr           Value of IYear(NYear)                                      0060
```

```
C  MinDif          Minimum spacing of key events.  For example, if          0061
C                  MinDif=2, then key events in both 1980 and 1981          0062
C                  would not be allowed.  This is used in the               0063
C                  Monte Carlo part of the technique, rather than to        0064
C                  check the real data.                                     0065
C  AStat           Statistic to use -- see above                            0066
C  ISign           Sign of key event: +1 for expected peak,                 0067
C                                     -1 for expected trough                0068
C ************************************************************************ 0069
C     *** Declarations ***                                                  0070
C                                                                          0071
      integer I, ILowYr, Int, IHiYr, J, K, NAbove, MinDif                  0072
      integer ISign(50), ABS, MSeed, Mod                                   0073
      integer Key(50), Key2(50), LWidth, RWidth                            0074
      double precision P(10000), TrialStat(10000), XX(100)                 0075
      double precision DataStat, Stat                                      0076
      double precision DFloat, Ran3                                        0077
      external Ran3, ReadData, CompT, CompD, CompW, WrtOut                 0078
      character*1 AStat                                                    0079
C                                                                          0080
      integer IYear(100), NYear, NKey, NSim, ISeed                         0081
C                                                                          0082
      common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,          0083
     &  ILowYr,IHiYr,LWidth,RWidth                                         0084
C ------------------------------------------------------------------------ 0085
C                                                                          0086
C     *** Read the data file                                               0087
C                                                                          0088
      Call ReadData(Key,ISign,AStat,MinDif,MSeed)                          0089
C                                                                          0090
C     *** Find the Test Statistic (T, D, or W) on the True Data            0091
C                                                                          0092
      IF (AStat .EQ. 'T') then                                             0093
         call CompT(ISign,Key,Stat)                                        0094
      ELSEIF (AStat .EQ. 'D') then                                         0095
         call CompD(ISign,Key,Stat)                                        0096
      ELSEIF (AStat .EQ. 'W') THEN                                         0097
         call CompW(ISign,Key,Stat)                                        0098
      ELSE                                                                 0099
         Print *, 'ERROR: Bad type of statistic: ',AStat                   0100
         stop                                                              0101
      ENDIF                                                                0102
      DataStat = Stat                                                      0103
C                                                                          0104
C                                                                          0105
C     *** Do NSim Monte Carlo Trials & Compute                             0106
C           Test Statistic for Each                                        0107
C                                                                          0108
      print *, '  Beginning simulations....'                               0109
C     First initialize the counter of larger P-values                      0110
      NAbove = 1                                                           0111
      Do 100 I=1, NSim                                                     0112
         if (mod(I,1000) .EQ. 0) print *,'  Simulation:',I                 0113
C        Pick out NKey years at random                                     0114
         Do 110 J=1, NKey                                                  0115
105         Key2(J) = ILowYr +                                             0116
     &           int( (IHiYr - ILowYr) * Ran3(iseed) + 0.5d0)              0117
C           If a year has missing data, choose another one                 0118
            IF (XX(Key2(J)-ILowYr+1) .LT. -9000.0d0) then                  0119
C              print *, 'Year with missing data replaced:', Key2(J)        0120
```

```
               goto 105                                                  0121
            ENDIF                                                        0122
C           If a year is too close to another year, choose another one   0123
            do 107, k=1,J-1                                              0124
            IF (ABS(Key2(J)-Key2(K)) .LT. MinDif) THEN                   0125
C              print *, 'Years too close:',(Key2(L),L=1,J)               0126
               goto 105                                                  0127
            ENDIF                                                        0128
107         continue                                                     0129
110      continue                                                        0130
C                                                                        0131
         IF (AStat .EQ. 'T') then                                        0132
            Call CompT(ISign,Key2,Stat)                                  0133
         ELSEIF (AStat .EQ. 'D') then                                    0134
            call CompD(ISign,Key2,Stat)                                  0135
         ELSEIF (AStat .EQ. 'W') then                                    0136
            call CompW(ISign,Key2,Stat)                                  0137
         ENDIF                                                           0138
C                                                                        0139
         TrialStat(I) = Stat                                             0140
         if (Stat .GT. DataStat) NAbove = NAbove + 1                     0141
C        Compute running P(i) for plotting                               0142
          P(I) = dfloat(NAbove) / dfloat(I+1)                            0143
100   continue                                                           0144
C                                                                        0145
C     *** Write Output & End                                             0146
C                                                                        0147
      Call WrtOut(DataStat,NAbove,Key,ISign,AStat,MinDif,MSeed)          0148
C                                                                        0149
      end                                                                0150
C                                                                        0151
C     *** End of main program (EPOCH)                                    0152
C ***************************************************************************0153
C ***************************************************************************0154
C                                                                        0155
      SUBROUTINE ReadData(Key,ISign,AStat,MinDif,MSeed)                  0156
C                                                                        0157
      integer I, ILowYr, IHiYr, Abs, MinDif                              0158
      integer IYear(100), ISign(50), Key(50), LWidth, RWidth             0159
      integer NYear, NKey, NSim, ISeed, MSeed, NSide, MWidth             0160
      double precision TrialStat(10000), P(10000), XX(100)               0161
      character*1 AStat                                                  0162
C                                                                        0163
      common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,        0164
     &  ILowYr,IHiYr,LWidth,RWidth                                       0165
C                                                                        0166
      open (unit=12, file='epoch.dat', status='OLD')                     0167
      read (12,*) NYear, NSim, NKey                                      0168
      do 110 i=1, NKey                                                   0169
         read (12,*) Key(i), ISign(i)                                    0170
         IF (ABS(ISign(i)) .NE. 1) then                                  0171
            print *, 'ERROR: Bad sign flag for key event #',i            0172
            print *, 'Value must be 1 or -1'                             0173
            STOP                                                         0174
         ENDIF                                                           0175
110   continue                                                           0176
      read (12,*) MinDif                                                 0177
      read (12,*) AStat                                                  0178
      read (12,*) MSeed                                                  0179
      read (12,*) NSide                                                  0180
```

```
      read (12,*) MWidth                                                0181
C                                                                       0182
C     *** Check for valid width                                         0183
C                                                                       0184
      if ((MWidth .LT. 1) .OR. (MWidth .GT. 4)) THEN                    0185
         print *,'ERROR: Incorrect epoch width: ',MWidth                0186
         print *,'Must be between 1 and 4.'                             0187
         stop                                                           0188
      ENDIF                                                             0189
C                                                                       0190
C     *** Determine the left and right widths on each side of the       0191
C         key events.                                                   0192
C                                                                       0193
      IF (NSide .eq. 2) THEN                                            0194
         LWidth = -MWidth                                               0195
         RWidth =  MWidth                                               0196
      ELSE IF (NSide .eq. 1) THEN                                       0197
         LWidth = -MWidth                                               0198
         RWidth = 0                                                     0199
      ELSE                                                              0200
         print *, 'ERROR: NSide equals ',NSide                          0201
         print *, 'NSide must equal 1 or 2 '                            0202
         STOP                                                           0203
      ENDIF                                                             0204
C                                                                       0205
C     *** Check that years are in order                                 0206
C                                                                       0207
      do 120 i=1, NYear                                                 0208
         read (12,*) IYear(i), XX(i)                                    0209
         IF ((I .GT.1) .AND. (IYear(i) .NE. IYear(i-1) +1)) THEN        0210
            print *,'ERROR: ID not sequential at ID',IYear(i)           0211
            stop                                                        0212
         ENDIF                                                          0213
120   continue                                                          0214
C                                                                       0215
C     *** Find first and last years in data, check key years            0216
C                                                                       0217
      ILowYr = IYear(1)                                                 0218
      IHiYr  = IYear(NYear)                                             0219
      Do 150 i = 1, NKey                                                0220
         IF ((Key(i) .GT. IHiYr) .OR. (Key(i) .LT. ILowYr)) THEN        0221
            PRINT *, 'ERROR: Key year out of range:', Key(i)            0222
            STOP                                                        0223
         ENDIF                                                          0224
150   continue                                                          0225
C                                                                       0226
C     *** Seed random number generator with a negative number.          0227
C         Use -ABS(SEED) from input file if it's not 0.  If seed        0228
C         is 0 , program stops and requests new seed.                   0229
C                                                                       0230
      IF (MSeed .GT. 0) THEN                                            0231
         MSeed = -1 * MSeed                                             0232
      ELSEIF (MSeed .EQ. 0) THEN                                        0233
         print *, 'ERROR: MSeed is 0. MSeed must be negative'           0234
         STOP                                                           0235
      ENDIF                                                             0236
      ISeed = MSeed                                                     0237
C                                                                       0238
C     Close file & return                                               0239
C                                                                       0240
```

```
      close (unit=12)                                                    0241
      return                                                             0242
      end                                                                0243
C ***************************************************************************0244
C                                                                        0245
      Subroutine CompT(ISign,Key,TStat)                                  0246
C                                                                        0247
C     Arguments:                                                         0248
C     Key -- (Input) An array of Key Events.  Key(i) is a value          0249
C            equal to the IYear value of the ith key event.              0250
C     ILowYear -- (Input) The first key year.                            0251
C     TStat  -- (Output) The returned value of the T-Statistic.          0252
C                                                                        0253
      Integer Key(50), I, I3, NB, JJ(50), I2, ILowYr, IHiYr              0254
      Integer IYear(100), ISign(50)                                      0255
      Double precision B, BMean, EMean, ESSD, BSSD, TStat                0256
      Double precision SD, dfloat, sqrt                                  0257
      double precision P(10000), TrialStat(10000), XX(100)               0258
      integer LWidth, RWidth, NYear, NKey, NSim, ISeed, K                0259
C                                                                        0260
      common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,        0261
     &  ILowYr,IHiYr,LWidth,RWidth                                       0262
C                                                                        0263
C     *** Put into the JJ array indexes to the XX array for the          0264
C         NKey key-event years.  The indexes are used below.             0265
C     *** While doing it, check for minus signs in input.                0266
C         They are not appropriate for the T-statistic.                  0267
C                                                                        0268
      DO 105 I = 1, NKey                                                 0269
         IF (ISign(i) .LT. 0) THEN                                       0270
            print *, 'ERROR:  "-1" flag input for key year',Key(i)       0271
            print *, 'Only "+1" flags are allowed for T-statistic.'      0272
            STOP                                                         0273
         ENDIF                                                           0274
         JJ(I) = Key(I) - ILowYr + 1                                     0275
105   CONTINUE                                                           0276
C                                                                        0277
C     *** Compute E-Mean (Mean of Event Values)                          0278
C                                                                        0279
      EMean = 0.0d0                                                      0280
      DO 110 I = 1, NKey                                                 0281
         EMean = EMean + XX(JJ(I))                                       0282
110   CONTINUE                                                           0283
      EMean = EMean / dfloat(NKey)                                       0284
C                                                                        0285
C     *** Compute ESSD (Sum of squared deviations for Event Col.)        0286
C                                                                        0287
      ESSD = 0.0d0                                                       0288
      DO 120 I = 1, NKey                                                 0289
         ESSD = ESSD + (XX(JJ(I)) - EMean) ** 2                          0290
120   CONTINUE                                                           0291
C                                                                        0292
C     *** Compute B-mean (Background)                                    0293
C                                                                        0294
      BMean = 0.0d0                                                      0295
      NB = 0                                                             0296
C     The first loop goes across the row                                 0297
      DO 140 I = LWidth, RWidth                                          0298
         if (I .EQ. 0) goto 140                                          0299
C        The inner loop goes down a column                               0300
```

```
                DO 150 I2 = 1, NKey                                                0301
C                  K indexes the XX array (1 ... NYear)                            0302
                   K = JJ(I2) + I                                                  0303
C                  Check for out-of-range background years                         0304
                   IF ((K .LT. 1) .OR. (K .GT. NYear)) THEN                        0305
                      B = -9999.0d0                                                0306
C                     print *,'Out of range year & row:',Key(I2)+I,K               0307
                   ELSE                                                            0308
                      B =  XX(K)                                                   0309
                   ENDIF                                                           0310
C                  Check for missing values ( = -9999)                             0311
                   IF (B .GT. -9000.0d0) THEN                                      0312
                      BMean = BMean + B                                            0313
                      NB = NB + 1                                                  0314
                   ENDIF                                                           0315
150             CONTINUE                                                           0316
140          CONTINUE                                                              0317
             BMean = BMean / dfloat(NB)                                            0318
C                                                                                  0319
C            *** Compute BSSD (Sum of squared deviations for Background Cols)       0320
C                                                                                  0321
             BSSD = 0.0d0                                                          0322
C            The first loop goes across the row                                    0323
             DO 160 I = LWidth, RWidth                                             0324
                IF (I .EQ. 0) goto 160                                             0325
C               Following loop goes down a column                                  0326
                DO 170 I2 = 1, NKey                                                0327
C                  K indexes the XX array (1 ... NYear)                            0328
                   K = JJ(I2) + I                                                  0329
C                  Check for out-of-range background years                         0330
                   IF ((K .LT. 1) .OR. (K .GT. NYear)) THEN                        0331
                      B = -9999.0d0                                                0332
C                     print *, 'Out of range year & row:',Key(I2)+I,K              0333
                   ELSE                                                            0334
                      B =  XX(K)                                                   0335
C                     Check for a background year that's on the list of key        0336
C                     years.  If it is, then make the data value missing.          0337
                      DO 130 I3 = 1 ,NKey                                          0338
                         IF (Key(I3) .EQ. IYear(K)) THEN                           0339
                            B = -9999.0d0                                          0340
C                           print *,'Background is also key',Key(I3),IYear(K)      0341
                         ENDIF                                                     0342
130                   continue                                                     0343
                   ENDIF                                                           0344
C                  Check for missing values ( = -9999)                             0345
                   IF (B .GT. -9000.0d0) BSSD = BSSD + (B - BMean) ** 2            0346
170             CONTINUE                                                           0347
160          CONTINUE                                                              0348
C                                                                                  0349
C            *** Compute Pooled Std Dev & T-Statistic                              0350
C                                                                                  0351
             SD = SQRT((ESSD+BSSD)*Dfloat(NB+NKey) /                               0352
            &   (DFloat(NKey * NB * (NB+NKey-2))))                                 0353
             TStat = (EMean - BMean) / SD                                          0354
C                                                                                  0355
C            *** Return                                                            0356
C                                                                                  0357
             RETURN                                                                0358
             END                                                                   0359
C                                                                                  0360
```

```
C ************************************************************************      0361
C                                                                               0362
      Subroutine CompD(ISign,Key,DStat)                                         0363
C                                                                               0364
C     Compute D-statistic.  D is the difference between the mean of             0365
C     values in event years and in background years.                            0366
C                                                                               0367
C     Arguments:                                                                0368
C     Key       (Input) An array of Key Events.  Key(i) is a value              0369
C               equal to the IYear value of the ith key event.                  0370
C     ILowYr    (Input) The first key year.                                     0371
C     DStat     (Output) The returned value of the D-Statistic.                 0372
C                                                                               0373
      Integer Key(50), I, NB, JJ(50), I2, I3, ILowYr, IHiYr                     0374
      Double precision B, BMean, EMean, DStat, DFloat                           0375
      double precision P(10000), TrialStat(10000), XX(100)                      0376
      integer IYear(100), ISign(50)                                             0377
      integer LWidth, RWidth, NYear, NKey, NSim, ISeed, K                       0378
C                                                                               0379
      common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,               0380
     &  ILowYr,IHiYr,LWidth,RWidth                                              0381
C                                                                               0382
C     *** Put into the JJ array indexes to the XX array for the                 0383
C         NKey key-event years.  The indexes are used below.                    0384
C     *** While doing it, check for minus signs in input.                       0385
C         They are not appropriate for the D-statistic.                         0386
C                                                                               0387
      DO 105 I = 1, NKey                                                        0388
         IF (ISign(i) .LT. 0) THEN                                              0389
            print *,'ERROR:  "-1" flag found at key year',Key(i)                0390
            print *, 'Only "+1" flags are allowed for D-statistic.'             0391
            STOP                                                                0392
         ENDIF                                                                  0393
         JJ(I) = Key(I) - ILowYr + 1                                            0394
105   CONTINUE                                                                  0395
C                                                                               0396
C     *** Compute E-Mean (Mean of Event Values)                                 0397
C                                                                               0398
      EMean = 0.0d0                                                             0399
      DO 110 I = 1, NKey                                                        0400
         EMean = EMean + XX(JJ(I))                                              0401
110   CONTINUE                                                                  0402
      EMean = EMean / dfloat(NKey)                                              0403
C                                                                               0404
C     *** Compute B-mean (Background)                                           0405
C                                                                               0406
      BMean = 0.0d0                                                             0407
      NB = 0                                                                    0408
C     The first loop goes across the row                                        0409
      DO 140 I = LWidth, RWidth                                                 0410
         if (I .EQ. 0) goto 140                                                 0411
C        The inner loop goes down a column                                      0412
         DO 150 I2 = 1, NKey                                                    0413
C           K indexes the XX array (1 ... NYear)                                0414
            K = JJ(I2) + I                                                      0415
C           Check for out-of-range background years                             0416
            IF ((K .LT. 1) .OR. (K .GT. NYear)) THEN                            0417
               B = -9999.0d0                                                    0418
C              print *,'Out of range year & row:',Key(I2)+I,K                   0419
            ELSE                                                                0420
```

```
                B =  XX(K)                                                      0421
C               Check for a background year that's on the list of key           0422
C               years.  If it is, then make the data value missing.             0423
                DO 120 I3 = 1, NKey                                             0424
                   IF (Key(I3) .EQ. IYear(K)) THEN                              0425
                      B = -9999.0d0                                             0426
C                     print *,'Background is also key',Key(I3),IYear(K)         0427
                   ENDIF                                                        0428
120             continue                                                        0429
              ENDIF                                                             0430
C             Check for missing values ( = -9999)                               0431
              IF (B .GT. -9000.0d0) THEN                                        0432
                 BMean = BMean + B                                              0433
                 NB = NB + 1                                                    0434
              ENDIF                                                             0435
150        CONTINUE                                                             0436
140     CONTINUE                                                                0437
        BMean = BMean / dfloat(NB)                                              0438
C                                                                               0439
C       *** Compute D-Statistic                                                 0440
C                                                                               0441
        DStat = EMean - BMean                                                   0442
C                                                                               0443
C       *** Return                                                              0444
C                                                                               0445
        RETURN                                                                  0446
        END                                                                     0447
C                                                                               0448
C ***************************************************************************** 0449
C                                                                               0450
        Subroutine CompW(ISign,Key,W)                                           0451
C                                                                               0452
C       Compute W-statistic.  W is analogous to a paired                        0453
C       Student's t statistic.                                                  0454
C                                                                               0455
C       Arguments:                                                              0456
C       Key       (Input) An array of Key Events.  Key(i) is a value            0457
C                 equal to the IYear value of the ith key event.                0458
C       ILowYr    (Input) The first key year.                                   0459
C       W         (Output) The returned value of the W-Statistic.               0460
C                                                                               0461
        Integer Key(50), I, NDif, JJ(50), I2, I3, ILowYr, IHiYr                 0462
        Double precision B, E, DifMean, DifVar, W                               0463
        Double precision dfloat, sqrt                                           0464
        Double precision Dif(100)                                               0465
        integer IYear(100), ISign(50)                                           0466
        integer LWidth, RWidth, NYear, NKey, NSim, ISeed, K                     0467
        double precision P(10000), TrialStat(10000), XX(100)                    0468
C                                                                               0469
        common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,             0470
       &  ILowYr,IHiYr,LWidth,RWidth                                            0471
C                                                                               0472
C       *** Put into the JJ array indexes to the XX array for the               0473
C           NKey key-event years.  The indexes are used below.                  0474
C                                                                               0475
        DO 105 I = 1, NKey                                                      0476
           JJ(I) = Key(I) - ILowYr + 1                                          0477
105     CONTINUE                                                                0478
C                                                                               0479
C       *** Compute differences between E and B and accumulate mean             0480
```

```
C                                                                       0481
      DifMean = 0.0d0                                                   0482
      NDif = 0                                                          0483
C     The first loop goes across the row of the epoch                   0484
      DO 140 I = LWidth, RWidth                                         0485
         if (I .EQ. 0) goto 140                                         0486
C        The inner loop goes down a column of the epoch                 0487
         DO 150 I2 = 1, NKey                                            0488
C           Get the "E" value for this row                              0489
            E = XX(JJ(I2))                                              0490
C           Get the "B" value for this comparison                       0491
C           Store the XX index for this B in K                          0492
            K = I + JJ(I2)                                              0493
C           Check for out-of-range background years                     0494
            IF ((K .LT. 1) .OR. (K .GT. NYear)) THEN                    0495
               B = -9999.0d0                                            0496
C              print *,'Out of range year & row:',Key(I2)+I,K           0497
            ELSE                                                        0498
               B =  XX(K)                                               0499
C              Check for a background year that's on the list of key    0500
C              years.  If it is, then make the data value missing.      0501
               DO 120 I3 = 1, NKey                                      0502
                  IF (Key(I3) .EQ. IYear(K)) THEN                       0503
                     B = -9999.0d0                                      0504
C                    print *,'Background is also key',Key(I3),IYear(K)  0505
                  ENDIF                                                 0506
120            continue                                                 0507
            ENDIF                                                       0508
C           Check for missing values ( = -9999)                         0509
            IF (B .GT. -9000.0d0) THEN                                  0510
               NDif = NDif + 1                                          0511
               Dif(NDif) = Dfloat(ISign(I2)) * (E - B)                  0512
               DifMean = DifMean + Dif(NDif)                            0513
            ENDIF                                                       0514
150      CONTINUE                                                       0515
140   CONTINUE                                                          0516
      DifMean = DifMean / DFloat(NDif)                                  0517
C                                                                       0518
C     *** Compute DifVar (Variance of differences)                      0519
C                                                                       0520
      DifVar = 0.0d0                                                    0521
      DO 160 I = 1, NDif                                                0522
         DifVar = DifVar + (Dif(I) - DifMean) ** 2                      0523
160   CONTINUE                                                          0524
      DifVar = DifVar / DFloat(NDif - 1)                                0525
C                                                                       0526
C     *** Compute W-Statistic by dividing mean difference by            0527
C         the std. error of the mean difference. See Snedecor           0528
C         and Cochran p. 85.                                            0529
C                                                                       0530
      W = DifMean / SQRT(DifVar/dfloat(NDif))                           0531
C                                                                       0532
C     *** Return                                                        0533
C                                                                       0534
      RETURN                                                            0535
      END                                                               0536
C                                                                       0537
C ***********************************************************************0538
C                                                                       0539
      SUBROUTINE WrtOut(DataStat,NAbove,Key,ISign,AStat,MinDif,MSeed)   0540
```

```
C                                                                       0541
      integer I, J, NAbove, Key(50), ILowYr, IHiYr, MOD                 0542
      integer LWidth, RWidth, MSeed, MinDif                             0543
      integer IYear(100), ISign(50), NYear, NKey, NSim, ISeed           0544
      double precision XX(100), P(10000), DataStat, TrialStat(10000)    0545
      character*1 AStat                                                 0546
C                                                                       0547
      common/bigblock/XX,IYear,NYear,NKey,NSim,ISeed,P,TrialStat,       0548
     &  ILowYr,IHiYr,LWidth,RWidth                                      0549
C                                                                       0550
      open (unit=13, file='epoch.out', status='UNKNOWN')                0551
C                                                                       0552
C     *** Write Header                                                  0553
C                                                                       0554
      write (13,*) ' Results of Superposed Epoch Analysis'              0555
      write (13,*) ' -------------------------------------'             0556
C                                                                       0557
C     *** Write Information on data                                     0558
C                                                                       0559
      I = RWidth - LWidth + 1                                           0560
      write  (13,501) I                                                 0561
501   format (/' Width of epoch: ',I8)                                 0562
      write  (13,502) -LWidth                                           0563
502   format (' Background Years :',/,8x,'The ',i1,' years on the Left') 0564
      write  (13,503) RWidth                                            0565
503   format (8x,'The ',i1,' years on the Right')                       0566
      write  (13,504) MSeed                                             0567
504   format (' Random number seed: ',I8)                               0568
      write  (13,505) NYear                                             0569
505   format (/' Number of rows in input data ..... ',I5)               0570
      write  (13,610) ILowYr                                            0571
610   format (' ID of first row ..... ',I5)                             0572
      write  (13,620) IHiYr                                             0573
620   format (' ID of last row ..... ',I5)                              0574
      write  (13,590) IHiYr-ILowYr+1                                    0575
590   format (' Range of ID values ..... ',I5)                          0576
C                                                                       0577
C     *** Report on Missing Values                                      0578
C                                                                       0579
      J = 0                                                             0580
      do 250 I = 1, NYear                                               0581
         if (XX(I) .LT. -9000) then                                     0582
            write (13,630) IYear(I)                                     0583
            J = J + 1                                                   0584
         ENDIF                                                          0585
250   continue                                                          0586
630   format ('    Missing data at ID # ..... ',I5)                     0587
      write (13,640) J                                                  0588
640   format (' Number of rows with missing data ..... ',I5)           0589
C                                                                       0590
C     *** Write Information on key years                                0591
C                                                                       0592
      write  (13,507) NKey                                              0593
507   format (/' Number of key periods: ',I4)                           0594
      write  (13,510) (I,Key(I),ISign(I), I=1,NKey)                     0595
510   format ('   Key period',1X,'#',I2,':',2X,I5,3X,'Sign:',SP,I4,S)   0596
C                                                                       0597
C     *** Write the statistics                                          0598
C                                                                       0599
      write (13,512) AStat                                              0600
```

```
512   format (/' Test statistic used was ..... ',A1)                    0601
      write  (13,515) DataStat                                           0602
515   format (/' Test statistic for real key events ..... ',F8.4)       0603
      write  (13,520) NSim                                               0604
520   format (/' Number of Monte Carlo trials ..... ',I5)               0605
      write  (13,522) MinDif                                             0606
522   format (' Minimum spacing between MC key events .... ',I5)        0607
      write  (13,525) AStat, NAbove                                      0608
525   format (' Number of trials with larger ',A1,' ..... ',I5)         0609
      write  (13,530) AStat, P(NSim)                                     0610
530   format (/' Estimated probability of a larger ',A1,'..... ',       0611
     &  F8.4)                                                            0612
      close(unit=13)                                                     0613
C                                                                        0614
C     *** Write out the P-vector                                         0615
C                                                                        0616
      open (unit=14,file='epoch.sig',status='unknown',blocksize=6000)    0617
C     Write Header                                                       0618
      write (14,*) ' Results from Superposed Epoch Analysis'             0619
      write (14,*) ' Series of P-Values from Monte Carlo Trial'          0620
      write (14,*) ' '                                                   0621
      write (14,*) ' '                                                   0622
      do 290 i=1,nsim                                                    0623
          if (mod(i,2) .eq. 0) write (14,540) I, TrialStat(I), P(I)      0624
290   continue                                                           0625
540   format (1x,I6,2(3x,F10.5))                                         0626
      close (unit=14)                                                    0627
C                                                                        0628
      return                                                             0629
      end                                                                0630
C                                                                        0631
C ************************************************************************0632
C ************************************************************************0633
C     Modified uniform random deviate generator. (C) 1986 by Numerical   0634
C     Recipes Software. Reproduced by permission, from the book          0635
C     Numerical Recipes: The Art of Scientific Computing, published      0636
C     by Cambridge University Press.                                     0637
C ************************************************************************0638
C                                                                        0639
      FUNCTION RAN3(IDUM)                                                0640
C                                                                        0641
      DOUBLE PRECISION fac, ran3                                         0642
      INTEGER mbig, mseed, mz, ma(55), iff, mj, iabs, mod, mk            0643
      INTEGER i, ii, k, inext, inextp, idum                              0644
      PARAMETER (MBIG=1000000000,MSEED=161803398,MZ=0,FAC=1.0d-9)        0645
      DATA IFF /0/                                                       0646
C                                                                        0647
      IF(IDUM.LT.0.OR.IFF.EQ.0)THEN                                      0648
        IFF=1                                                            0649
        MJ=MSEED-IABS(IDUM)                                              0650
        MJ=MOD(MJ,MBIG)                                                  0651
        MA(55)=MJ                                                        0652
        MK=1                                                             0653
        DO 11 I=1,54                                                     0654
          II=MOD(21*I,55)                                                0655
          MA(II)=MK                                                      0656
          MK=MJ-MK                                                       0657
          IF(MK.LT.MZ)MK=MK+MBIG                                         0658
          MJ=MA(II)                                                      0659
11      CONTINUE                                                         0660
```

```
          DO 13 K=1,4                                                          0661
            DO 12 I=1,55                                                       0662
              MA(I)=MA(I)-MA(1+MOD(I+30,55))                                   0663
              IF(MA(I).LT.MZ)MA(I)=MA(I)+MBIG                                  0664
   12       CONTINUE                                                           0665
   13     CONTINUE                                                             0666
          INEXT=0                                                              0667
          INEXTP=31                                                            0668
          IDUM=1                                                               0669
        ENDIF                                                                  0670
        INEXT=INEXT+1                                                          0671
        IF(INEXT.EQ.56)INEXT=1                                                 0672
        INEXTP=INEXTP+1                                                        0673
        IF(INEXTP.EQ.56)INEXTP=1                                               0674
        MJ=MA(INEXT)-MA(INEXTP)                                                0675
        IF(MJ.LT.MZ)MJ=MJ+MBIG                                                 0676
        MA(INEXT)=MJ                                                           0677
        RAN3=MJ*FAC                                                            0678
        RETURN                                                                 0679
        END                                                                    0680
C                                                                              0681
C ***************************************************************************  0682
C End of EPOCH.FOR                                                             0683
C ***************************************************************************  0684
```

Appendix III - Program EPOWER
sample input file (EPOWER.DAT) with annotations

35	**Number of years in each simulated data set.**
3	**Number of key event years in each data set.**
1.5d0	**Amount of difference in mean recruitment between background and key event years.**
0.0d0	**Time slope of recruitment.**
-72365	**Negative seed for RAN3.**
2	**Epower definition: 1 for background years to the left. 2 for left & right.**
2	**Number of background years preceding (and possibly following) key events.**
10	**Number of data sets generated (outer trials).**
10000	**Number of Monte Carlo trials per data set (inner trials).**
0.2d0	**Autoregressive parameter.**

Appendix IV - Program EPOWER Listing

```
C ************************************************************************      0001
C     EPOWER.FOR                                                               0002
C     Examine the power of superposed epoch analysis.                          0003
C ************************************************************************      0004
      PROGRAM EPOWER                                                           0005
C     Prager/Hoenig/Payton  November, 1989                                     0006
C ------------------------------------------------------------------------      0007
C     Input Data File Structure (Free Format):                                 0008
C     ----------------------------------------                                 0009
C     Line 1:     NYear    Number of years in each simulated data set.         0010
C     Line 2:     NKey     Number of key event years in each data set.         0011
C     Line 3:     Delta    Amount of difference in mean recruitment            0012
C                          between background and key event years.             0013
C     Line 4:     Slope    Time slope of recruitment.                          0014
C     Line 5:     MSeed    Negative seed for RAN3.                             0015
C     Line 6:     NSide    Sides to be used for MWidth.                        0016
C                          '1':  One sided. Uses left side.                    0017
C                          '2':  Two sided.                                    0018
C     Line 7:     MWidth   Half-width of epoch on each side of                 0019
C                          key event if two sided, full-width if one           0020
C                          sided.                                              0021
C     Line 8:     NData    Number of data sets generated (outer trials).       0022
C     Line 9:     NMonte   Number of Monte Carlo trials (inner trials).        0023
C     Line 10:    ARparm   Autoregressive parameter.                           0024
C ========================================================================      0025
C  File Unit   DDname          Action   Contents                               0026
C  ---------   ----------      ------   ----------------------------           0027
C  Unit 12     EPOWER.DAT      read     Input parameters                       0028
C  Unit 13     EPOWER.OUT      write    Main output file                       0029
C  Unit 14     EPOWER.DET      write    Detailed output file                   0030
C ========================================================================      0031
C  ***  Major variables  ***                                                   0032
C                                                                              0033
C  XX(i)            Recruitment in year i.                                     0034
C  NYear            Number of years in each simulated data set.                0035
C  NKey             Number of key events in each simulated data set.           0036
C  NData            Number of phony data sets to be evaluated.                 0037
C  Key(j)           Year (index) of key event j (j = 1 ... NKey)               0038
C  Key2()           Same, in Monte Carlo simulation.                           0039
C  NMonte           Number of Monte Carlo trials for randomization.            0040
C  ARparm           Autoregressive parameter.                                  0041
C  ARFlag           Switch used to determine which way to generate time        0042
C                   series                                                     0043
C                                                                              0044
C  StatD            Difference statistic.                                      0045
C  PD               Prob > D, determined by randomization                      0046
C                                                                              0047
C  StatT            Student's t statistic (= Prager/Hoenig "T" statistic)      0048
C  PRT              Prob > t, determined by randomization                      0049
C  PTT              Prob > t, determined from tables                           0050
C                                                                              0051
C  StatW            Quasi-paired t, as used in first paper by Hoenig &         0052
C                   Prager                                                     0053
C  PW               Prob > W, determined by randomization                      0054
C                                                                              0055
C  StatQ            Student's paired t statistic (classic)                     0056
C  PRQ              Prob > Q, determined by randomization.                     0057
C  PTQ              Prob > Q, determined from tables.                          0058
C                                                                              0059
C  NTSigT           (e.g.; others similar) # of instances with PTT < alpha     0060
```

```
C  PTSigT          NTSigT / NData                                          0061
C  MSeed           Original random number seed                             0062
C  ISeed           Running random number seed                              0063
C ***********************************************************************  0064
C     *** Declarations ***                                                 0065
C                                                                          0066
      integer  Key(10), IData, MSeed                                       0067
      integer NSigD, NRSigT, NTSigT, NSigW, NTSigQ, NRSigQ                 0068
      logical xmonte                                                       0069
      double precision alpha, aninv,xi                                     0070
      double precision StatD, StatT, StatW, StatQ                          0071
      double precision PD, PRT, PTT, PW, PRQ, PTQ                          0072
      double precision PSigD, PRSigT, PTSigT, PSigW, PRSigQ, PTSigQ        0073
C                                                                          0074
C ***********************************************************************  0075
C     Common declarations and statement for EPOWER.FOR                     0076
C ***********************************************************************  0077
C                                                                          0078
      integer ndata, nmonte, nyear, nkey, iseed,                           0079
     +    mindif, arflag, LWidth, RWidth, MWidth                           0080
      double precision xx(50), delta, slope, arparm                        0081
                                                                           0082
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,            0083
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth         0084
C                                                                          0085
C ***********************************************************************  0086
C                                                                          0087
      common/flags/xmonte                                                  0088
      MinDif = 1                                                           0089
      Call ReadData(MSeed)                                                 0090
      alpha = 0.05d0                                                       0091
      nsigd  = 0                                                           0092
      ntsigt = 0                                                           0093
      nrsigt = 0                                                           0094
      nsigw  = 0                                                           0095
      ntsigq = 0                                                           0096
      nrsigq = 0                                                           0097
      xmonte = .TRUE.                                                      0098
      print *,'Power study beginning....'                                  0099
C                                                                          0100
C     *** Run the analysis for each of NDATA fake data sets                0101
C                                                                          0102
      Do 1001 IData = 1, NData                                             0103
         xi = dfloat(IData)/10.0d0                                         0104
         if (int(xi) .eq. xi) print *,'IData = ',Idata                     0105
C                                                                          0106
C        *** Generate a data set:                                          0107
C                                                                          0108
         if (ARFlag .eq. 0) then                                           0109
            call MDataL(Key)                                               0110
         else if (ARFlag .eq. 1) then                                      0111
            call MDataAR(Key)                                              0112
         else                                                              0113
            print *, 'Bad value for ARFlag.'                               0114
            stop                                                           0115
         endif                                                             0116
C                                                                          0117
C        *** Compute the statistics: D, T, W, and Paired t (Pt):           0118
C                                                                          0119
         Call CompD(Key,StatD)                                             0120
```

```
          Call CompT(Key,StatT,PTT)                                          0121
          Call CompW(Key,StatW)                                              0122
          Call CompQ(Key,StatQ,PTQ)                                          0123
C                                                                            0124
C         *** Compute significance probabilities of D, T, W, Q by            0125
C         randomization                                                      0126
C                                                                            0127
          xmonte = .FALSE.                                                   0128
          Call ranprob(statd,statt,statw,statq,PD,PRT,PW,PRQ)                0129
          xmonte = .TRUE.                                                    0130
C                                                                            0131
C         *** Increment counters of significant statistics                   0132
C                                                                            0133
          if (pd  .lt. alpha) nsigd  = nsigd  + 1                            0134
          if (prt .lt. alpha) nrsigt = nrsigt + 1                            0135
          if (ptt .lt. alpha) ntsigt = ntsigt + 1                            0136
          if (pw  .lt. alpha) nsigw  = nsigw  + 1                            0137
          if (prq .lt. alpha) nrsigq = nrsigq + 1                            0138
          if (ptq .lt. alpha) ntsigq = ntsigq + 1                            0139
C                                                                            0140
          call wrtout(key,idata,statd,statt,statw,statq,                     0141
     +       pd,prt,ptt,pw,prq,ptq,mseed)                                    0142
C                                                                            0143
1001  continue                                                               0144
C                                                                            0145
C     *** Compute proportion of cases found significant                      0146
C                                                                            0147
      ANINV = 1.0d0 / dfloat(NData)                                          0148
      PSigD  = dfloat(NSigD)  * ANINV                                        0149
      PRSigT = dfloat(NRSigT) * ANINV                                        0150
      PTSigT = dfloat(NTSigT) * ANINV                                        0151
      PSigW  = dfloat(NSigW)  * ANINV                                        0152
      PRSigQ = dfloat(NRSigQ) * ANINV                                        0153
      PTSigQ = dfloat(NTSigQ) * ANINV                                        0154
C                                                                            0155
C                                                                            0156
C     *** Call the summary output routine to print a summary of results      0157
C                                                                            0158
         Call WrtSum(NSigD, NRSigT, NTSigT, NSigW, NRSigQ, NTSigQ,           0159
     +        PSigD, PRSigT, PTSigT, PSigW, PRSigQ, PTSigQ)                  0160
                                                                             0161
C                                                                            0162
C     *** End of main program (EPOWER)                                       0163
      end                                                                    0164
C                                                                            0165
C ***************************************************************************0166
C ***************************************************************************0167
C                                                                            0168
      SUBROUTINE ReadData(MSeed)                                             0169
C                                                                            0170
      integer MSeed, NSide                                                   0171
C                                                                            0172
C                                                                            0173
C ***************************************************************************0174
C     Common declarations and statement for EPOWER.FOR                       0175
C ***************************************************************************0176
C                                                                            0177
      integer ndata, nmonte, nyear, nkey, iseed,                             0178
     +    mindif, arflag, LWidth, RWidth, MWidth                             0179
      double precision xx(50), delta, slope, arparm                          0180
```

```
                                                                        0181
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,         0182
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth      0183
C                                                                       0184
C ***********************************************************************0185
C                                                                       0186
      open (unit=12, file='EPOWER.DAT', status='old')                   0187
      read (12,*) NYear                                                 0188
      read (12,*) NKey                                                  0189
      read (12,*) Delta                                                 0190
      read (12,*) Slope                                                 0191
      read (12,*) MSeed                                                 0192
      read (12,*) NSide                                                 0193
      read (12,*) MWidth                                                0194
      read (12,*) NData                                                 0195
      read (12,*) NMonte                                                0196
      read (12,*) ARparm                                                0197
C                                                                       0198
C     *** Seed random number generator with a negative number.          0199
C         Use -ABS(SEED) from input file if it's not 0.  Use a          0200
C         fixed number (-85625) if given seed is 0.                     0201
C                                                                       0202
      IF (MSeed .LT. 0) THEN                                            0203
         ISeed = MSeed                                                  0204
      ELSE IF (MSeed .GT. 0) THEN                                       0205
         ISeed = -1 * MSeed                                             0206
      ELSE IF (MSeed .EQ. 0) THEN                                       0207
         MSeed = -815625                                                0208
         ISeed = MSeed                                                  0209
      END IF                                                            0210
C                                                                       0211
C     *** Determine the left and right widths on each side of the       0212
C         key events.                                                   0213
C                                                                       0214
      if (NSide .eq. 2) then                                            0215
         LWidth = -MWidth                                               0216
         RWidth = MWidth                                                0217
      else if (NSide .eq. 1) then                                       0218
         LWidth = -MWidth                                               0219
         RWidth = 0                                                     0220
      else                                                              0221
         print *, 'ERROR: NSide equals ',NSide                          0222
         print *, 'NSide must equal 1 or 2 '                            0223
         stop                                                           0224
      endif                                                             0225
C                                                                       0226
C     *** Set ARFlag (Flag for autoregression)                          0227
C                                                                       0228
      if (abs(ARparm) .lt. 1.0d-10) then                                0229
         arflag = 0                                                     0230
      else                                                              0231
         arflag = 1                                                     0232
      end if                                                            0233
C                                                                       0234
      close (unit=12)                                                   0235
      return                                                            0236
      end                                                               0237
C                                                                       0238
C ***********************************************************************0239
C ***********************************************************************0240
```

```
C                                                                       0241
      SUBROUTINE MDataL(Key)                                            0242
C                                                                       0243
C     Generate a random dataset for the epoch power test.               0244
C     This generates data for a linear (non-autoregressive) model.      0245
C                                                                       0246
      double precision GasDev, Ran3                                     0247
      integer  JPick(50),I,J,Key(10)                                    0248
C                                                                       0249
C                                                                       0250
C ************************************************************************0251
C     Common declarations and statement for EPOWER.FOR                  0252
C ************************************************************************0253
C                                                                       0254
      integer ndata, nmonte, nyear, nkey, iseed,                        0255
     +    mindif, arflag, LWidth, RWidth, MWidth                        0256
      double precision xx(50), delta, slope, arparm                     0257
                                                                        0258
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,         0259
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth      0260
C                                                                       0261
C ************************************************************************0262
C                                                                       0263
C                                                                       0264
C     JPick = 0 if a background, 1 if a key event year.                 0265
C     Delta = amount by which key years differ from background.         0266
C     NKey  = number of key event years                                 0267
C     XX    = recruitment                                               0268
C                                                                       0269
      do 100 i = 1, NYear                                               0270
C        * Generate normal random recruitments for each year            0271
         XX(i) = gasdev(ISeed) + Slope * i                              0272
         JPick(i) = 0                                                   0273
100   continue                                                          0274
C                                                                       0275
C      *** Pick NKey key event years, and make sure they are unique     0276
C                                                                       0277
      do 150 j = 1, NKey                                                0278
120      Key(j) = 1 + int(dfloat(nyear) * RAN3(iseed))                  0279
         IF (JPick(key(j)) .eq. 1) THEN                                 0280
            goto 120                                                    0281
          ELSE                                                          0282
            JPick(Key(j)) = 1                                           0283
            XX(Key(j)) = XX(Key(j)) + Delta                             0284
          ENDIF                                                         0285
150   continue                                                          0286
      return                                                            0287
      end                                                               0288
C                                                                       0289
C ************************************************************************0290
C ************************************************************************0291
C                                                                       0292
      Subroutine MDataAR(Key)                                           0293
C                                                                       0294
C     This subroutine generates the time series of observations based on0295
C     Y(t) = ARparm * Y(T-1) + Delta * Z(T) + E(T)                      0296
C                                                                       0297
C WHERE   Y(T)    = OBSERVATION IN YEAR T.                              0298
C         Y(T-1) = OBSERVATION IN YEAR T-1.                             0299
C         ARparm     = CONSTANT                                         0300
```

```
C         Delta  = CONSTANT                                                    0301
C         Z(T)   = 0, YEAR T IS NOT A KEY EVENT YEAR                           0302
C                  1, YEAR T IS A KEY EVENT YEAR                              0303
C         E(T)   = RANDOM ERROR TERM ( N(0,1) )                                0304
C                                                                              0305
      double precision GasDev, Ran3, e(50),x1, x2, probke, arq                 0306
      integer  JPick(50),I,J,Key(10), limit                                    0307
C                                                                              0308
C                                                                              0309
C ***************************************************************************  0310
C     Common declarations and statement for EPOWER.FOR                         0311
C ***************************************************************************  0312
C                                                                              0313
      integer ndata, nmonte, nyear, nkey, iseed,                               0314
     +    mindif, arflag, LWidth, RWidth, MWidth                               0315
      double precision xx(50), delta, slope, arparm                            0316
                                                                               0317
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,                0318
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth             0319
C                                                                              0320
C ***************************************************************************  0321
C                                                                              0322
C     JPick = 0 if a background, 1 if a key event year.                        0323
C     Delta = amount by which key years differ from background.                0324
C     NKey  = number of key event years                                        0325
C     XX    = recruitment                                                      0326
C                                                                              0327
C     First generate set of key event years.                                   0328
C     Set marker for key event years to zero\and genearate error term          0329
                                                                               0330
      arq = 1.0d0 - ARParm                                                     0331
      do 100 i = 1, NYear                                                      0332
         JPick(i) = 0                                                          0333
*        e(i) = gasdev(ISeed) * arq                                            0334
         e(i) = gasdev(ISeed)                                                  0335
100   continue                                                                 0336
C                                                                              0337
C      *** Pick NKey key event years, and make sure they are unique            0338
C                                                                              0339
      do 150 j = 1, NKey                                                       0340
120      Key(j) = 1 + int(dfloat(nyear) * RAN3(iseed))                         0341
         IF (JPick(key(j)) .eq. 1) THEN                                        0342
            goto 120                                                           0343
          ELSE                                                                 0344
            JPick(Key(j)) = 1                                                  0345
          ENDIF                                                                0346
150   continue                                                                 0347
C                                                                              0348
C                                                                              0349
C     Run the random series for 100 to 300 times before using the values       0350
C                                                                              0351
C     probke is the probability of a key event                                 0352
      probke = nyear / nkey                                                    0353
      x1 = 0.0d0                                                               0354
      limit = 100 + int(200.0d0 * ran3(iseed))                                 0355
      do 160 i = 1, limit                                                      0356
*        x2 = arparm * x1 + gasdev(iseed) * arq                                0357
         x2 = arparm * x1 + gasdev(iseed)                                      0358
         if (ran3(iseed) .gt. probke) x2 = x2 + delta                          0359
         x1 = x2                                                               0360
```

```
160   continue                                                          0361
C                                                                       0362
C     Now generate an autoregressive time series based on the formula.  0363
C                                                                       0364
      xx(1) =  arparm * x1 + delta * dfloat(jpick(1)) + e(1)            0365
      if (jpick(1) .eq. 1) xx(1) = xx(1) + delta                        0366
      do 170 i = 2, nyear                                               0367
         xx(i) = arparm * xx(i-1) + delta * dfloat(jpick(i)) + e(i)     0368
170   continue                                                          0369
C                                                                       0370
      return                                                            0371
      end                                                               0372
C                                                                       0373
C ***********************************************************************0374
C ***********************************************************************0375
C                                                                       0376
      SUBROUTINE ranprob(d,t,w,q,pd,prt,pw,prq)                         0377
C                                                                       0378
C     Find significance of statistics by randomization.                 0379
C                                                                       0380
      integer nd, nt, nw, nq, key2(10), i, j, k                         0381
      double precision pd, prt, pw, prq, dummy, anm1                    0382
      double precision d, t, w, q, dd, tt, ww, qq                       0383
      double precision dx, tx, wx, qx                                   0384
      double precision abs, ran3                                        0385
C                                                                       0386
C                                                                       0387
C                                                                       0388
C ***********************************************************************0389
C     Common declarations and statement for EPOWER.FOR                  0390
C ***********************************************************************0391
C                                                                       0392
      integer ndata, nmonte, nyear, nkey, iseed,                        0393
     +    mindif, arflag, LWidth, RWidth, MWidth                        0394
      double precision xx(50), delta, slope, arparm                     0395
                                                                        0396
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,         0397
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth      0398
C                                                                       0399
C ***********************************************************************0400
C                                                                       0401
C        * Initialize the counters for the number of trials with        0402
C          larger values of each statistic                              0403
C                                                                       0404
         ND = 1                                                         0405
         NT = 1                                                         0406
         NW = 1                                                         0407
         NQ = 1                                                         0408
C                                                                       0409
         dd = d                                                         0410
         tt = t                                                         0411
         ww = w                                                         0412
         qq = q                                                         0413
C                                                                       0414
C        *** Run NMONTE randomization trials                            0415
                                                                        0416
         Do 100 I=1, NMonte                                             0417
C                                                                       0418
C           *** Pick out NKey years at random                           0419
C                                                                       0420
```

```
                do 110 j = 1, nkey                                              0421
105               key2(j) = 1 + int(dfloat(nyear) * ran3(iseed))                0422
C                                                                               0423
C                 *** If too close to another key year, choose another one      0424
C                                                                               0425
                  do 107, k=1,J-1                                               0426
                     if (abs(Key2(J)-Key2(K)) .LT. MinDif) goto 105             0427
107               continue                                                      0428
110             continue                                                        0429
C                                                                               0430
                Call CompD(Key2,DX)                                             0431
                Call CompT(Key2,TX,dummy)                                       0432
                Call CompW(Key2,WX)                                             0433
                Call CompQ(Key2,QX,dummy)                                       0434
                if (DX .GT. DD) ND = ND + 1                                     0435
                if (TX .GT. TT) NT = NT + 1                                     0436
                if (WX .GT. WW) NW = NW + 1                                     0437
                if (QX .GT. QQ) NQ = NQ + 1                                     0438
100           continue                                                          0439
C                                                                               0440
C             *** Compute significance probabilities for all statistics         0441
C                                                                               0442
              ANM1  = 1.0d0 / dfloat(NMonte + 1)                                0443
C                                                                               0444
              PD    = dfloat(ND) * ANM1                                         0445
              PRT   = dfloat(NT) * ANM1                                         0446
              PW    = dfloat(NW) * ANM1                                         0447
              PRQ   = dfloat(NQ) * ANM1                                         0448
C                                                                               0449
              return                                                            0450
              end                                                               0451
C                                                                               0452
C ******************************************************************************0453
C ******************************************************************************0454
C                                                                               0455
      Subroutine CompD(Key,DStat)                                               0456
C                                                                               0457
C     Compute D-statistic.  D is the difference between the mean of             0458
C     values in event years and in background years                             0459
C                                                                               0460
C     Key(i)(Input)  The indexes of the nkey key years.                         0461
C     DStat (Output) The returned value of the D-Statistic.                     0462
C                                                                               0463
      Integer I, J, K, NB, Key(10), bflag(50), kflag(50)                        0464
      Double precision bbar, ebar, DStat                                        0465
C                                                                               0466
C                                                                               0467
C                                                                               0468
C ******************************************************************************0469
C     Common declarations and statement for EPOWER.FOR                          0470
C ******************************************************************************0471
C                                                                               0472
      integer ndata, nmonte, nyear, nkey, iseed,                                0473
     +    mindif, arflag, LWidth, RWidth, MWidth                                0474
      double precision xx(50), delta, slope, arparm                             0475
                                                                                0476
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,                 0477
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth              0478
C                                                                               0479
C ******************************************************************************0480
```

```
C                                                                       0481
C     *** Initialize flags to zero                                      0482
C                                                                       0483
      do 100 i = 1, nyear                                               0484
         kflag(i) = 0                                                   0485
         bflag(i) = 0                                                   0486
100   continue                                                          0487
C                                                                       0488
C     *** Compute ebar (mean of key event recruitments)                 0489
C                                                                       0490
      ebar = 0.0d0                                                      0491
      do 110 i = 1, nkey                                                0492
         ebar = ebar + xx(key(i))                                       0493
         kflag(key(i)) = 1                                              0494
110   continue                                                          0495
      ebar = ebar / dfloat(nkey)                                        0496
C                                                                       0497
C     *** Flag background years.  Since epochs may overlap, they are    0498
C         accumulated into a sum only after all are flagged.            0499
C                                                                       0500
      do 130 j = 1, nkey                                                0501
         do 220 i = lwidth, rwidth                                      0502
            if (i .eq. 0) goto 220                                      0503
C           * store the index for this year temporarily in k            0504
            k = key(j) + i                                              0505
            if ((k .ge. 1) .and. (k .lt. nyear)) then                   0506
               if (kflag(k) .eq. 0) bflag(k) = 1                        0507
            endif                                                       0508
220      continue                                                       0509
130   continue                                                          0510
C                                                                       0511
C     *** Sum the background years in bbar and add up NB.               0512
C                                                                       0513
      nb = 0                                                            0514
      bbar = 0.0d0                                                      0515
      do 140 i = 1, nyear                                               0516
         if (bflag(i) .eq. 1) then                                      0517
            nb = nb + 1                                                 0518
            bbar = bbar + xx(i)                                         0519
         end if                                                         0520
140   continue                                                          0521
      bbar = bbar / dfloat(nb)                                          0522
C                                                                       0523
      dstat = ebar - bbar                                               0524
C                                                                       0525
      return                                                            0526
      end                                                               0527
C                                                                       0528
C ***********************************************************************0529
C ***********************************************************************0530
C                                                                       0531
      SUBROUTINE COMPQ(Key,stat,Prob)                                   0532
C                                                                       0533
C     Compute a paired Student's t-statistic and its significance       0534
C     probability.  This is done by preparing two data sets (DATA1      0535
C     and DATA2) of sample size N (= NKey), and then calling            0536
C     subroutine TPTEST from Numerical Recipes.                         0537
C                                                                       0538
C     stat  (Output) The returned value of the paired t-statistic.      0539
C     Prob  (Output) The returned probability of a greater paired t.    0540
```

```
C                                                                       0541
      integer i, j, k, n, nb, key(10), kflag(50), jLWidth, jRWidth      0542
      double precision data1(50), data2(50), stat, prob                 0543
C                                                                       0544
C                                                                       0545
C                                                                       0546
C ***********************************************************************0547
C     Common declarations and statement for EPOWER.FOR                  0548
C ***********************************************************************0549
C                                                                       0550
      integer ndata, nmonte, nyear, nkey, iseed,                        0551
     +    mindif, arflag, LWidth, RWidth, MWidth                        0552
      double precision xx(50), delta, slope, arparm                     0553
                                                                        0554
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,         0555
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth      0556
C                                                                       0557
C ***********************************************************************0558
C                                                                       0559
C     Initialize sample size and key year status flags.                 0560
C                                                                       0561
      n = nkey                                                          0562
      do 100 i = 1, nyear                                               0563
         kflag(i) = 0                                                   0564
100   continue                                                          0565
C                                                                       0566
C     Copy key year recruitment values into the DATA1 array             0567
C     and set all key year status flags                                 0568
C                                                                       0569
      do 120 i = 1, nkey                                                0570
         k = key(i)                                                     0571
         data1(i) = xx(k)                                               0572
         kflag(k) = 1                                                   0573
120   continue                                                          0574
C                                                                       0575
C     Find the mean value of background years for each key event.       0576
C                                                                       0577
      do 200 j = 1, nkey                                                0578
         jLWidth = LWidth                                               0579
         jRWidth = RWidth                                               0580
         data2(j) = 0.0d0                                               0581
         nb = 0                                                         0582
140      do 220 i = jLWidth, jRWidth                                    0583
C           * Don't use the key event year itself.                      0584
            if (i .eq. 0) goto 220                                      0585
C           * Store the xx index for this year in k                     0586
            k = key(j) + i                                              0587
C           * If the year is off the end of the series, don't use it.   0588
            if ((k .lt. 1) .or. (k .gt. nyear)) goto 220                0589
C           * If the year isn't a(nother) key year, add it in.          0590
            if (kflag(k) .eq. 0) then                                   0591
               data2(j) = data2(j) + xx(k)                              0592
               nb = nb + 1                                              0593
            endif                                                       0594
220      continue                                                       0595
C                                                                       0596
C        * Make sure there are some background years for the test.      0597
C                                                                       0598
         if (nb .eq. 0) then                                            0599
C           print *, 'No background years for epoch #',j                0600
```

```
C           print *, 'Half-epoch widened by 1.'                          0601
            if (RWidth .ne. 0) then                                      0602
               jRWidth = RWidth + 1                                      0603
               jLWidth = LWidth - 1                                      0604
            else                                                         0605
               jLWidth = LWidth - 1                                      0606
            endif                                                        0607
            goto 140                                                     0608
         endif                                                           0609
C                                                                        0610
C        * Compute the mean background recruitment for this key event    0611
C                                                                        0612
         data2(j) = data2(j) / DFLOAT(nb)                                0613
200   continue                                                           0614
C                                                                        0615
C     *** Compute the test statistic and significance probability        0616
C                                                                        0617
      call tptest(data1,data2,n,stat,prob)                               0618
C                                                                        0619
      return                                                             0620
      end                                                                0621
C                                                                        0622
C ************************************************************************0623
C ************************************************************************0624
C                                                                        0625
      SUBROUTINE COMPT(Key,stat,Prob)                                    0626
C                                                                        0627
C     Compute Student's t-statistic and its significance probability.    0628
C     This is done by preparing two data sets (DATA1 and DATA2) and      0629
C     then calling subroutine TTEST from Numerical Recipes.              0630
C                                                                        0631
C     stat  (Output) The returned value of the t-Statistic.              0632
C     Prob  (Output) The returned probability of a greater t.            0633
C                                                                        0634
      Integer I, J, k, N1, N2, Key(10), KFlag(50), BFlag(50)             0635
      Double precision DATA1(50), DATA2(50), stat, Prob                  0636
C                                                                        0637
C                                                                        0638
C                                                                        0639
C ************************************************************************0640
C     Common declarations and statement for EPOWER.FOR                   0641
C ************************************************************************0642
C                                                                        0643
      integer ndata, nmonte, nyear, nkey, iseed,                         0644
     +    mindif, arflag, LWidth, RWidth, MWidth                         0645
      double precision xx(50), delta, slope, arparm                      0646
                                                                         0647
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,          0648
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth       0649
C                                                                        0650
C ************************************************************************0651
C                                                                        0652
C     Initialize flags and counters to zero.                             0653
C                                                                        0654
      N1 = NKey                                                          0655
      do 100 i = 1, NYear                                                0656
         kflag(i) = 0                                                    0657
         bflag(i) = 0                                                    0658
100   continue                                                           0659
C                                                                        0660
```

```
C     Copy key year recruitment values into the DATA1 array.
C
      do 110 i = 1, NKey
         DATA1(i) = xx(key(i))
         kflag(key(i)) = 1
110   continue
C
C     Flag background years.  Since epochs may overlap, they are
C     transferred to DATA2 only after all are flagged.
C
      do 130 j = 1, nkey
         do 220 i = lwidth, rwidth
            if (i .eq. 0) goto 220
C           Store the index for this year in k
            k = key(j) + i
            if ((k .ge. 1) .and. (k .lt. nyear)) then
               if (kflag(k) .eq. 0) bflag(k) = 1
            endif
220      continue
130   continue
C
C     Transfer the background years to DATA2 and accumulate N2.
C
      N2 = 0
      do 140 i = 1, nyear
         if (bflag(i) .eq. 1) then
            n2 = n2 + 1
            data2(n2) = xx(i)
         end if
140   continue
C
      call ttest(data1,data2,stat,prob,n1,n2)
C
      RETURN
      END
C
C ***************************************************************************
C ***************************************************************************
C
      Subroutine CompW(Key,W)
C
C     Compute W-statistic, analogous to a paired Student's t-statistic.
C
C     W (Output):  The returned value of the W-Statistic.
C
      Integer I, I2, I3, K, NDif, Key(10)
      Double precision B, E, DifMean, DifVar, W
      Double precision dfloat, sqrt, Dif(100)
C
C
C ***************************************************************************
C     Common declarations and statement for EPOWER.FOR
C ***************************************************************************
C
      integer ndata, nmonte, nyear, nkey, iseed,
     +    mindif, arflag, LWidth, RWidth, MWidth
      double precision xx(50), delta, slope, arparm

      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth
```

```
C                                                                       0721
C ***********************************************************************0722
C                                                                       0723
C     *** Compute differences between E and B and accumulate mean       0724
C                                                                       0725
      DifMean = 0.0d0                                                   0726
      NDif = 0                                                          0727
C     The first loop goes across the row of the epoch                   0728
      DO 140 I = LWidth, RWidth                                         0729
         if (I .EQ. 0) goto 140                                         0730
C        The inner loop goes down a column of the epoch                 0731
         DO 150 I2 = 1, NKey                                            0732
C           Get the "E" value for this row                              0733
            E = XX(Key(I2))                                             0734
C           Get the "B" value for this comparison                       0735
C           Store the XX index for this B in K                          0736
            K = Key(I2) + I                                             0737
C           Check for out-of-range background years                     0738
            IF ((K .LT. 1) .OR. (K .GT. NYear)) THEN                    0739
               B = -9999.0d0                                            0740
C              print *,'Out of range year & row:',Key(I2)+I,K           0741
            ELSE                                                        0742
               B =  XX(K)                                               0743
C              Check for a background year that's on the list of key    0744
C              years.  If it is, then make the data value missing.      0745
               do 110 I3=1,NKey                                         0746
                  IF (Key(I3) .EQ. K) B = -9999.0d0                     0747
110            continue                                                 0748
            end if                                                      0749
C           * Check for missing values ( = -9999)                       0750
            if (B .GT. -9000.0d0) then                                  0751
               NDif = NDif + 1                                          0752
               Dif(NDif) = E - B                                        0753
               DifMean = DifMean + Dif(NDif)                            0754
            endif                                                       0755
150      continue                                                       0756
140   continue                                                          0757
      DifMean = DifMean / DFloat(NDif)                                  0758
C                                                                       0759
C     *** Compute DifVar (Variance of differences)                      0760
C                                                                       0761
      DifVar = 0.0d0                                                    0762
      do 160 I = 1, NDif                                                0763
         DifVar = DifVar + (Dif(I) - DifMean) **2                       0764
160   continue                                                          0765
      DifVar = DifVar / DFloat(NDif - 1)                                0766
C                                                                       0767
C     *** Compute W-Statistic by dividing mean difference by            0768
C         the std. error of the mean difference. See Snedecor           0769
C         and Cochran p. 85.                                            0770
C                                                                       0771
      W = DifMean / SQRT(DifVar/dfloat(NDif))                           0772
C                                                                       0773
      RETURN                                                            0774
      END                                                               0775
C                                                                       0776
C ***********************************************************************0777
C ***********************************************************************0778
C                                                                       0779
      SUBROUTINE WrtOut(Key,IData,StatD,StatT,StatW,StatQ,              0780
```

```
     +      PD,PRT,PTT,PW,PRQ,PTQ,MSeed)                                    0781
                                                                             0782
C                                                                            0783
C     Write output for epoch power study                                     0784
C                                                                            0785
      integer I, J, IData, Key(10), ifile, MSeed                             0786
      character*10 filedd                                                    0787
      character*15 atype                                                     0788
      double precision StatD,StatT,StatW,StatQ                               0789
      double precision PD,PRT,PTT,PW,PRQ,PTQ                                 0790
C                                                                            0791
C                                                                            0792
C ***************************************************************************0793
C     Common declarations and statement for EPOWER.FOR                       0794
C ***************************************************************************0795
C                                                                            0796
      integer ndata, nmonte, nyear, nkey, iseed,                             0797
     +    mindif, arflag, LWidth, RWidth, MWidth                             0798
      double precision xx(50), delta, slope, arparm                          0799
                                                                             0800
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,              0801
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth           0802
C                                                                            0803
C ***************************************************************************0804
C                                                                            0805
C     *** The first time through, open files & write headers                 0806
C                                                                            0807
      If (IData .EQ. 1) THEN                                                 0808
C                                                                            0809
C        * Open files & write headers                                        0810
         do 100 ifile = 13, 14                                               0811
            if (ifile .eq. 13) filedd = 'EPOWER.OUT'                         0812
            if (ifile .eq. 14) filedd = 'EPOWER.DET'                         0813
            open (unit=ifile, file=filedd, status='unknown')                 0814
C                                                                            0815
            write (ifile,400)                                                0816
400         format (' Power Analysis of Superposed Epoch Analysis')          0817
            write (ifile,410)                                                0818
410         format (' ----------------------------------------')             0819
C                                                                            0820
C           Compute variables needed for output only                         0821
C                                                                            0822
C           Full width of epoch                                              0823
            I = RWidth - LWidth + 1                                          0824
            if (ARFlag .eq. 1) then                                          0825
                  atype = ' Autoregressive'                                  0826
               else                                                          0827
                  atype = '.. Uncorrelated'                                  0828
            end if                                                           0829
C                                                                            0830
C           * Write input data to output file                                0831
C                                                                            0832
            write  (ifile, 510) NData                                        0833
510         format (/' Number of simulated data sets ..... ',I5)            0834
            write  (ifile, 515) NYear                                        0835
515         format ( ' Number of years in each data set ..... ',I5)         0836
            write  (ifile, 520) NKey                                         0837
520         format ( ' Number of key event years ..... ',I5)                0838
            write  (ifile, 525) I                                            0839
525         format (' Width of epoch..... ',I5)                             0840
            write  (ifile, 526) -LWidth
```

```
526         format (' Background Years :',/,8x,'The ',i1,                    0841
     +               ' years on the Left')                                   0842
            write  (ifile, 527) RWidth                                       0843
527         format (8x,'The ',i1,' years on the Right')                      0844
            write  (ifile, 532) atype                                        0845
532         format (/' Model type .....................',A15)                0846
            write  (ifile, 535) Delta                                        0847
535         format (' Delta recruitment in key event years ..... ',F5.3)     0848
            write  (ifile, 540) Slope                                        0849
540         format (' Slope of recruitment time trend ..... ',F5.3)          0850
            if (ARFlag .eq. 1) write (ifile, 542) ARparm                     0851
542         format (' Autoregressive parameter ................. ',F5.3)     0852
            write  (ifile, 545) NMonte                                       0853
545         format (/' Number of M.C. trials in randomization ... ',I5)      0854
            write  (ifile, 550) MSeed                                        0855
550         format (' Random number seed ..................... ',I7)         0856
C                                                                            0857
C           * Write header for statistics                                    0858
C                                                                            0859
            if (ifile .eq. 14) write (ifile,560)                             0860
560         format (//' Trial    D      t      ',                            0861
     +       'W      Q   P(>D) Pr(>t) Pt(>t)  P(>W) Pr(>Q) Pt(>Q)'/)         0862
100      continue                                                            0863
      END IF                                                                 0864
C                                                                            0865
C     The following is to avoid a warning message that Key isn't used:       0866
      J = Key(1)                                                             0867
C                                                                            0868
C     *** Write the statistics                                               0869
C                                                                            0870
      write  (14,600) IData, StatD, StatT, StatW, StatQ,                     0871
     +   PD, PRT, PTT, PW, PRQ, PTQ                                          0872
600   format (1x,I3,1X, 4(F6.3,1X),6(1X,F6.3))                              0873
C                                                                            0874
      return                                                                 0875
      end                                                                    0876
C                                                                            0877
C ***************************************************************************0878
C ***************************************************************************0879
C                                                                            0880
      SUBROUTINE WrtSum(NSigD, NRSigT, NTSigT, NSigW, NRSigQ, NTSigQ,        0881
     +    PSigD, PRSigT, PTSigT, PSigW, PRSigQ, PTSigQ)                      0882
C                                                                            0883
C     Write output summary for epoch power study                             0884
C                                                                            0885
      integer          NSigD, NRSigT, NTSigT, NSigW, NRSigQ, NTSigQ          0886
      double precision PSigD, PRSigT, PTSigT, PSigW, PRSigQ, PTSigQ          0887
      double precision AN2, UD, URT, UTT, UW, URQ, UTQ                       0888
      double precision     LD, LRT, LTT, LW, LRQ, LTQ                        0889
      double precision     SD, SRT, STT, SW, SRQ, STQ                        0890
C                                                                            0891
C                                                                            0892
C ***************************************************************************0893
C     Common declarations and statement for EPOWER.FOR                       0894
C ***************************************************************************0895
C                                                                            0896
      integer ndata, nmonte, nyear, nkey, iseed,                             0897
     +    mindif, arflag, LWidth, RWidth, MWidth                             0898
      double precision xx(50), delta, slope, arparm                          0899
                                                                             0900
```

```
      common/bigblock/xx, delta, slope, mindif, lwidth, rwidth,          0901
     +   ndata, nmonte, nyear, nkey, iseed, arflag, arparm, mwidth       0902
                                                                         0903
C ***********************************************************************0904
C                                                                        0905
C                                                                        0906
C     * Write Header                                                     0907
      write (13,500)                                                     0908
500   format (//' SUMMARY OF RESULTS:')                                  0909
      write (13,510)                                                     0910
510   format (' -------------------')                                    0911
C                                                                        0912
C     *** Compute upper & lower confidence bounds on powers:             0913
      AN2 = 2.0d0 / dsqrt(dfloat(Ndata))                                 0914
C                                                                        0915
      SD = AN2 * dsqrt(PSigD*(1.0d0-PSigD))                              0916
      LD = PSigD - SD                                                    0917
      UD = PSigD + SD                                                    0918
C                                                                        0919
      STT = AN2 * dsqrt(PTSigT*(1.0d0-PtSigT))                           0920
      LTT = PTSigT - STT                                                 0921
      UTT = PTSigT + STT                                                 0922
C                                                                        0923
      SRT = AN2 * dsqrt(PRSigT*(1.0d0-PRSigT))                           0924
      LRT = PRSigT - SRT                                                 0925
      URT = PRSigT + SRT                                                 0926
C                                                                        0927
      SW = AN2 * dsqrt(PSigW*(1.0d0-PSigW))                              0928
      LW = PSigW - SW                                                    0929
      UW = PSigW + SW                                                    0930
C                                                                        0931
      SRQ = AN2 * dsqrt(PRSigQ*(1.0d0-PRSigQ))                           0932
      LRQ = PRSigQ - SRQ                                                 0933
      URQ = PRSigQ + SRQ                                                 0934
C                                                                        0935
      STQ = AN2 * dsqrt(PTSigQ*(1.0d0-PTSigQ))                           0936
      LTQ = PTSigQ - STQ                                                 0937
      UTQ = PTSigQ + STQ                                                 0938
C                                                                        0939
C     *** Write the statistics                                           0940
520   format(/' Statistic:',t20,'D',t26,'T(r)',t35,'T(t)',t47,'W',       0941
     +   t53,'Q(r)',t62,'Q(t)')                                          0942
600   format (/' N Significant:',  t15, 6(I6,3X))                        0943
610   format (/' Est. Power:',     t15, 6(F6.3,3X))                      0944
620   format (/' Lower 95% CL:',   t15, 6(F6.3,3X))                      0945
630   format ( ' Upper 95% CL:',   t15, 6(F6.3,3X))                      0946
      write  (13,520)                                                    0947
      write  (13,600) NSigD, NRSigT, NTSigT, NSigW, NRSigQ, NTSigQ       0948
      write  (13,610) PSigD, PRSigT, PTSigT, PSigW, PRSigQ, PTSigQ       0949
      write  (13,620) LD, LRT, LTT, LW, LRQ, LTQ                         0950
      write  (13,630) UD, URT, UTT, UW, URQ, UTQ                         0951
C                                                                        0952
      write (13,700)                                                     0953
      write (13,705)                                                     0954
      write (13,710)                                                     0955
      write (13,720)                                                     0956
      write (13,730)                                                     0957
      write (13,740)                                                     0958
      write (13,750)                                                     0959
      write (13,760)                                                     0960
C
```

```
700   format (//' Notes on Statistics:')                                0961
705   format (' -------------------')                                   0962
710   format (' D is the difference statistic.')                        0963
720   format (' T(t) and T(r) are Student''s t-statistics, with')       0964
730   format ('  prob. from tables / computed by randomization.')       0965
740   format (' W is the paired t-like statistic of Prager & Hoenig.')  0966
750   format (' Q(t) and Q(r) are paired Student''s t-stats, with')     0967
760   format ('  prob. from tables / computed by randomization.')       0968
C                                                                       0969
      close(unit=13)                                                    0970
      close(unit=14)                                                    0971
C                                                                       0972
      return                                                            0973
      end                                                               0974
C ***********************************************************************0975
      FUNCTION RAN3(IDUM)                                               0976
C                                                                       0977
C ***********************************************************************0978
C     Modified Uniform random deviate generator. (C) 1986 by Numerical  0979
C     Recipes Software.  Reproduced by permission, from the book        0980
C     Numerical Recipes: The Art of Scientific Computing, published by  0981
C     Cambridge University Press.                                       0982
C ***********************************************************************0983
C                                                                       0984
      DOUBLE PRECISION fac, ran3                                        0985
      INTEGER mbig, mseed, mz, ma(55), iff, mj, iabs, mod, mk           0986
      INTEGER i, ii, k, inext, inextp, idum                             0987
      PARAMETER (MBIG=1000000000,MSEED=161803398,MZ=0,FAC=1.0d-9)       0988
      DATA IFF /0/                                                      0989
C                                                                       0990
      IF(IDUM .LT. 0 .OR. IFF .EQ. 0)THEN                               0991
        IFF = 1                                                         0992
        MJ = MSEED - IABS(IDUM)                                         0993
        MJ = MOD(MJ,MBIG)                                               0994
        MA(55) = MJ                                                     0995
        MK = 1                                                          0996
        DO 11 I=1,54                                                    0997
          II=MOD(21*I,55)                                               0998
          MA(II)=MK                                                     0999
          MK=MJ-MK                                                      1000
          IF(MK.LT.MZ)MK=MK+MBIG                                        1001
          MJ=MA(II)                                                     1002
11      CONTINUE                                                        1003
        DO 13 K=1,4                                                     1004
          DO 12 I=1,55                                                  1005
            MA(I)=MA(I)-MA(1+MOD(I+30,55))                              1006
            IF(MA(I).LT.MZ)MA(I)=MA(I)+MBIG                             1007
12        CONTINUE                                                      1008
13      CONTINUE                                                        1009
        INEXT=0                                                         1010
        INEXTP=31                                                       1011
        IDUM=1                                                          1012
      ENDIF                                                             1013
      INEXT=INEXT+1                                                     1014
      IF(INEXT.EQ.56)INEXT=1                                            1015
      INEXTP=INEXTP+1                                                   1016
      IF(INEXTP.EQ.56)INEXTP=1                                          1017
      MJ=MA(INEXT)-MA(INEXTP)                                           1018
      IF(MJ.LT.MZ)MJ=MJ+MBIG                                            1019
      MA(INEXT)=MJ                                                      1020
```

```
      RAN3=MJ*FAC                                                        1021
      RETURN                                                             1022
      END                                                                1023
C                                                                        1024
C ***********************************************************************1025
      FUNCTION GASDEV(IDUM)                                              1026
C                                                                        1027
C ***********************************************************************1028
C     Modified Gaussian random deviate generator. (C) 1986 by Numerical  1029
C     Recipes Software.  Reproduced by permission, from the book         1030
C     Numerical Recipes: The Art of Scientific Computing, published by   1031
C     Cambridge University Press.                                        1032
C ***********************************************************************1033
C                                                                        1034
      integer iset, idum                                                 1035
      double precision fac,gset,gasdev,r,ran3,v1,v2                      1036
      double precision sqrt, log                                         1037
C                                                                        1038
      DATA ISET/0/                                                       1039
      IF (ISET .EQ. 0) THEN                                              1040
1        V1 = 2.0d0 * RAN3(IDUM) - 1.0d0                                 1041
         V2 = 2.0d0 *RAN3(IDUM) - 1.0d0                                  1042
         R = V1**2 + V2**2                                               1043
         IF (R .GE. 1.0d0) GO TO 1                                       1044
         FAC = SQRT(-2.0d0 * LOG(R) / R)                                 1045
         GSET = V1 * FAC                                                 1046
         GASDEV = V2 * FAC                                               1047
         ISET = 1                                                        1048
      ELSE                                                               1049
        GASDEV = GSET                                                    1050
        ISET = 0                                                         1051
      ENDIF                                                              1052
      RETURN                                                             1053
      END                                                                1054
C                                                                        1055
C ***********************************************************************1056
C ***********************************************************************1057
C                                                                        1058
      SUBROUTINE AVEVAR(DATA,N,AVE,VAR)                                  1059
C                                                                        1060
C ***********************************************************************1061
C     Modified averaged variance subroutine. (C) 1986 by Numerical       1062
C     Recipes Software.  Reproduced by permission, from the book         1063
C     Numerical Recipes: The Art of Scientific Computing, published by   1064
C     Cambridge University Press. Called by TTEST.                       1065
C ***********************************************************************1066
C                                                                        1067
      double precision data(50), s, ave, var                             1068
      integer j, n                                                       1069
C                                                                        1070
      ave=0.0                                                            1071
      var=0.0                                                            1072
      do 11 j=1,n                                                        1073
        ave=ave+data(j)                                                  1074
11    continue                                                           1075
      ave=ave/n                                                          1076
      do 12 j=1,n                                                        1077
        s=data(j)-ave                                                    1078
        var=var+s*s                                                      1079
12    continue                                                           1080
```

```
      var=var/dfloat(n-1)                                                1081
      return                                                             1082
      end                                                                1083
C                                                                        1084
C ***********************************************************************1085
C ***********************************************************************1086
C                                                                        1087
      SUBROUTINE TTEST(DATA1,DATA2,T,PROB,N1,N2)                         1088
C                                                                        1089
C ***********************************************************************1090
C     Modified Student's t-test subroutine. (C) 1986 by Numerical        1091
C     Recipes Software.  Reproduced by permission, from the book         1092
C     Numerical Recipes: The Art of Scientific Computing, published by   1093
C     Cambridge University Press.                                        1094
C ***********************************************************************1095
C                                                                        1096
      double precision data1(50),data2(50)                               1097
      double precision ave1, ave2, var, var1, var2, t, prob              1098
      double precision betai                                             1099
      integer df, n1, n2                                                 1100
      logical xmonte                                                     1101
      common/flags/xmonte                                                1102
C                                                                        1103
      call avevar(data1,n1,ave1,var1)                                    1104
      call avevar(data2,n2,ave2,var2)                                    1105
      df=n1+n2-2                                                         1106
      var=((n1-1)*var1+(n2-1)*var2)/df                                   1107
      t=(ave1-ave2)/sqrt(var*(1.0d0/n1+1.0d0/n2))                        1108
C     Note: prob. adjusted for one-tailed test.                          1109
C     Compute only if randomizing                                        1110
C     This is a one-tailed test                                          1111
      if (xmonte) then                                                   1112
         if (t .gt. 0.0d0) then                                          1113
            prob=0.5d0*betai(0.5d0*df,0.5d0,df/(df+t**2))                1114
         else                                                            1115
            prob = 1.0d0                                                 1116
         endif                                                           1117
      endif                                                              1118
      return                                                             1119
      end                                                                1120
C                                                                        1121
C ***********************************************************************1122
C ***********************************************************************1123
C                                                                        1124
      SUBROUTINE TPTEST(DATA1,DATA2,N,T,PROB)                            1125
C                                                                        1126
C ***********************************************************************1127
C     Modified paired Student's t-test subroutine. (C) 1986 by           1128
C     Numerical Recipes Software.  Reproduced by permission, from the    1129
C     book Numerical Recipes: The Art of Scientific Computing,           1130
C     published by Cambridge University Press.                           1131
C ***********************************************************************1132
C                                                                        1133
      double precision DATA1(50),DATA2(50)                               1134
      double precision sqrt, betai, sd                                   1135
      double precision ave1, ave2, var1, var2, cov, t, prob              1136
      integer n, j, df                                                   1137
      logical xmonte                                                     1138
      common/flags/xmonte                                                1139
C                                                                        1140
```

```
      CALL AVEVAR(DATA1,N,AVE1,VAR1)                                    1141
      CALL AVEVAR(DATA2,N,AVE2,VAR2)                                    1142
      COV = 0.0d0                                                       1143
      DO 11 J = 1, N                                                    1144
        COV = COV + (DATA1(J) - AVE1) * (DATA2(J) - AVE2)               1145
11    CONTINUE                                                          1146
      DF=N-1                                                            1147
      COV=COV/DF                                                        1148
      SD = SQRT((VAR1 + VAR2 - 2.0d0 * COV) / N)                        1149
      T = (AVE1 - AVE2) / SD                                            1150
      if (xmonte) then                                                  1151
         if (t .gt. 0.0d0) then                                         1152
            prob=0.5d0*betai(0.5d0*df,0.5d0,df/(df+t**2))               1153
         else                                                           1154
            prob = 1.0d0                                                1155
         endif                                                          1156
      endif                                                             1157
      RETURN                                                            1158
      END                                                               1159
C                                                                       1160
C ***********************************************************************1161
      FUNCTION BETAI(A,B,X)                                             1162
C                                                                       1163
C ***********************************************************************1164
C     Modified incomplete beta function. (C) 1986 by Numerical Recipes  1165
C     Software.  Reproduced by permission, from the book Numerical      1166
C     Recipes: The Art of Scientific Computing, published by            1167
C     Cambridge University Press.                                       1168
C***********************************************************************1169
C                                                                       1170
      double precision a, b, x, bt, betai, betacf, dlog, gammln         1171
C                                                                       1172
      if(x .lt. 0.0d0 .or. x .gt. 1.0d0) pause 'bad argument x in betai'1173
      if(x .eq. 0.0d0 .or. x .eq. 1.0d0 ) then                          1174
        bt = 0.0d0                                                      1175
      else                                                              1176
        bt = exp(gammln(a+b)-gammln(a)-gammln(b)                        1177
     &      +a*dlog(x)+b*dlog(1.0d0-x))                                 1178
      endif                                                             1179
      if(x.lt.(a+1.0d0)/(a+b+2.0d0))then                                1180
        betai=bt*betacf(a,b,x)/a                                        1181
        return                                                          1182
      else                                                              1183
        betai=1.0d0-bt*betacf(b,a,1.0d0-x)/b                            1184
        return                                                          1185
      endif                                                             1186
      end                                                               1187
C                                                                       1188
C***********************************************************************1189
      FUNCTION BETACF(A,B,X)                                            1190
C                                                                       1191
C***********************************************************************1192
C     Modified continued fraction for incomplete beta function.         1193
C     (C) 1986 by Numerical Recipes Software.  Reproduced by            1194
C     permission, from the book Numerical Recipes: The Art of           1195
C     Scientific Computing, published by Cambridge University Press.    1196
C***********************************************************************1197
C                                                                       1198
      double precision am, bm, az, qab, qap, qam, bz, ap, bp, app, bpp  1199
      double precision a, b, x, em, tem, betacf, eps, d, aold           1200
```

```
      integer m, itmax                                                  1201
      parameter (itmax=100,eps=3.0d-7)                                  1202
C                                                                       1203
      am=1.0d0                                                          1204
      bm=1.0d0                                                          1205
      az=1.0d0                                                          1206
      qab=a+b                                                           1207
      qap=a+1.0d0                                                       1208
      qam=a-1.0d0                                                       1209
      bz=1.0d0-qab*x/qap                                                1210
      do 11 m=1,itmax                                                   1211
        em=m                                                            1212
        tem=em+em                                                       1213
        d=em*(b-m)*x/((qam+tem)*(a+tem))                                1214
        ap=az+d*am                                                      1215
        bp=bz+d*bm                                                      1216
        d=-(a+em)*(qab+em)*x/((a+tem)*(qap+tem))                        1217
        app=ap+d*az                                                     1218
        bpp=bp+d*bz                                                     1219
        aold=az                                                         1220
        am=ap/bpp                                                       1221
        bm=bp/bpp                                                       1222
        az=app/bpp                                                      1223
        bz=1.                                                           1224
        if (abs(az - aold) .lt. eps * abs(az)) go to 1                  1225
11    continue                                                          1226
      pause 'a or b too big, or itmax too small'                        1227
1     betacf=az                                                         1228
      return                                                            1229
      end                                                               1230
C                                                                       1231
C*******************************************************************   1232
      FUNCTION GAMMLN(XX)                                               1233
C                                                                       1234
C*******************************************************************   1235
C     Modified log gamma function. (C) 1986 by Numerical Recipes        1236
C     Software.  Reproduced by permission, from the book Numerical      1237
C     Recipes: The Art of Scientific Computing, published by            1238
C     Cambridge University Press.                                       1239
C*******************************************************************   1240
C                                                                       1241
      double precision COF(6),STP,HALF,ONE,FPF,X,TMP,SER,xx, gammln     1242
      integer j                                                         1243
C                                                                       1244
      data cof,stp/76.18009173d0,-86.50532033d0,24.01409822d0,          1245
     *    -1.231739516d0,.120858003d-2,-.536382d-5,2.50662827465d0/     1246
C                                                                       1247
      data half,one,fpf/0.5d0,1.0d0,5.5d0/                              1248
C                                                                       1249
      x = xx-one                                                        1250
      tmp = x+fpf                                                       1251
      tmp=(x+half)*log(tmp)-tmp                                         1252
      ser=one                                                           1253
      do 11 j=1,6                                                       1254
        x=x+one                                                         1255
        ser=ser+cof(j)/x                                                1256
11    continue                                                          1257
      gammln=tmp+log(stp*ser)                                           1258
      return                                                            1259
      end                                                               1260
```

```
C*************************************************************************      1261
C ************************************************************************      1262
C                                                                               1263
C End of EPOWER.FOR                                                             1264
C ************************************************************************
```